

Debreceni Egyetem

Informatikai Kar

Távoktatás, e-learning

**Témavezető:**

dr. Sztrik János

egyetemi tanár

Készítette:

Ignác József

informatika tanár

Debrecen, 2010

TARTALOMJEGYZÉK

Bevezetés.....	3
Az atipikus tanulás szükségessége	4
Távoktatás és e-learning	6
Távoktatás	6
E-learning.....	11
Szerepek	11
Szerepproblémák.....	14
A távoktatás tananyaga	17
Hogyan kezdjünk hozzá?	19
Témaválasztás, célcsoport.....	20
Előmunkálatok.....	21
Hogyan tanuljak?	22
Célkitűzés	24
A tananyag.....	26
Eljárások és függvények.....	28
Fejezetlapok.....	28
Tartalom oldal	36
Alapok.....	36
Haladó megjelenítés	43
Feladatok	47
Feladatbank	62
Összefoglalás	63
Irodalomjegyzék.....	64
Hivatkozások.....	64

BEVEZETÉS

Szakdolgozat. Szak-dolgozat... Szakmai dolgozat? Morzsolgatom magamban a szavakat, próbálom összefüggésbe hozni a megjelenés formáját (szakmai dolgozat) a témával. Úgy írok szakmai dolgozatot, hogy tudom, a téma inkább kifelé visz a szakmából, mint erősítené azt.

Elgondolkodtunk már azon, milyen feladatai vannak egy tanárnak? Bemenni az órára, „leadni” az anyagot, esetenként számon kérni...? Vagy valami más? Ha egy apró történelmi visszatekintést teszünk, akkor láthatjuk, hogy a régi időkben a tanító (tanár) szakma tekintélynek örvendett. Tisztelet övezte, felnéztek a tudására. Nem csak a személyét tisztelték, hanem magát a szakmát is: a tudást, bölcsességet, önfegyelmet, szorgalmat, szigort, hiszen ezek elengedhetetlen tulajdonságai voltak egy igazi tanítónak, tanárnak. Ezzel összhangban a diákság is egészen más volt. Lehet, hogy a félelem is közrejátszott, de nem mertek ellentmondani a tanárnak, sokkal szigorúbban vették a szabályokat, mivel erre tanították őket. A tanár alkalomadtán meg is fenýíthette a csíntalan diákot, tehát ez a nevelő szerepére is utal. Összefoglalva a tanár nemcsak oktatott, hanem nevelt, mert az iskola elsődleges célja az életre való felkészítés volt, nem pedig az oktatás.

S mi történik most? Teljesen felbomlott ez a kapcsolatrendszer. A tanár tekintélye erősen csökkent, a diákok jogai megnőttek. Persze, mondhatjuk, hogy felhígult a szakma, ami igaz is. Olyan oktatók kerültek be az intézményekbe, akiknek a szakmai tudása és/vagy az emberi értékei kívánnivalót hagynak maguk után. Egyre többet hallani olyan szituációt, hogy csak azért jelentkezett valaki tanárképzőbe, mert máshová nem vették fel, tehát csupán a diploma utáni kényszer miatt lesz valaki pedagógus, nem pedig elhivatottságból teszi azt. Éppen ezért a társadalom már nem tekinti olyan értékesnek a „tanárlétet”, mint anno, s bizony, joggal. Természetesen vannak más aspektusok is, amit érdemes figyelembe venni, mielőtt valaki levonná a konzekvenciákat, s ráadásul a dolgozatom témája sem az, hogy a fennálló rendet véleményezzem.

Miért hoztam mégis elő ezt az apró történelmi visszatekintést? Csupán azért, mert az oktatás folyamatosan változik. Úgy tűnik, egyre kevesebb nevelő funkciója van, s egyre több a tényanyag átadásának kényszere. A távoktatás, az e-learning bizony azok közé tartozik, amelyek csak erősítik ezt az irányvonalat. Egyre kevesebb a kontaktus a tanár és a diák között, s a tananyaggal próbáljuk kompenzálni a nevelési elveket. Vajon sikerül? Lehet-e helyettesíteni a tanárt egy számítógéppel, vagy egy offline tananyaggal? Miért van mégis szükségünk ezekre az atipikus oktatási/tanulási formákra? Hogyan tudjuk úgy elkészíteni őket, hogy a célcsoportok szívesen használják őket? Erről szólnak a következő oldalak.

AZ ATÍPIKUS TANULÁS SZÜKSÉGESSÉGE

A szokásos tanulási rend szerint a gyermek belép az általános iskolába 6, esetleg 7 évesen. Nyolc év alatt elvégzi az általános iskolát, majd a középiskola négy éve következik. Ezek után – amennyiben az eredményei megfelelőek – főiskolai, egyetemi évek következnek. Tizenegynehány év nappali, iskolarendszerű képzésben eltöltött idő. De mi történik ezek után? Az immár felnőtt ember elmegy dolgozni, s a következő évek a családról, a munkáról szólnak...?

Túl gyorsan változik körülöttünk a világ ahhoz, hogy a fenti ideális rendet követni tudjuk. A munkaerőpiaci igények folyamatos változása, a munkanélküliség, a megszerzett tudás elavulása arra indíttat, hogy ne elégedjünk meg az ifjúkorban megszerzett tudással, hanem életünk további szakaszaiban is próbáljunk meg használatos, eladható tudást szerezni. S máris elérkeztünk egy manapság igen sokszor használt fogalomhoz, az *élethosszig tartó tanulás* fogalmához. Az egész életen át tartó tanulás a születéstől az idős korig terjedő egységes folyamat, amely magában foglalja a tanulás összes formáját. Lépést kell tartanunk a változásokkal, frissítenünk kell tudásunkat. Ehhez az *atipikus képzési és tanulási formák* nyújtják a legjobb lehetőségeket, hiszen az iskolarendszerben végzett képzések utáni tanulási formák már nem elsősorban a megszokott keretek között történnek, inkább levelező, esti tagozaton, iskolarendszeren kívül; tanfolyami keretek adnak lehetőséget a továbbfejlődésre. Célja, hogy lehetővé tegye azon ismeretek és készségek elsajátítását, amelyek segítségével az egyén aktív részese tud lenni a társadalomnak, képes az életminőségének növelésére, a hatékony probléma-megoldásra, a csoportmunkára, és rendelkezik azokkal a motivációs készségekkel, amelyek segítségével megtalálja helyét a munkaerőpiacon is.

Nem elég csak a célról beszélni, fontos tényező a célcsoport is. Sokszor elfeledjük, hogy a jelenlegi tudástársadalomban a legfontosabb termelési tényező maga az ember, még akkor is, ha esetleg a hátrányos helyzetűek csoportjába sorolható az egyén. Egy ország, nemzet versenyképessége bizony a rendelkezésre álló humán erőforrások minőségétől függ. Az atipikus képzések ezért különösen indokoltak olyan célcsoportok számára, mint amikor a mozgásukban korlátozott emberekről van szó, vagy amikor a fizikailag vagy szociálisan nehezen mobilizálhatókról beszélünk, vagy ha a GYES-en, GYED-en lévő anyák, mondjuk településükön nem férnek hozzá a képzésekhez, mert nincs, és nagyobb távolságra utazni pedig nem tudnak, például a gyermekelhelyezési problémák miatt. Természetesen még számos érv szól az atipikus tanulási módszerek mellett, különösen az e-learning mellett. Nyugodtan mondhatjuk azonban, hogy megfelelő tananyag kifejlesztésével bármely

társadalmi réteg számára közvetíthető a szakmák és foglalkozások elsajátításához szükséges ismeretek döntő többsége. Jelenleg mintegy ötévente megkétszereződik a rendelkezésünkre álló tudásanyag tömege. Összefoglalhatjuk, hogy minden állampolgárnak szüksége van ismeretek, készségek és kompetenciák elsajátítására és folyamatos frissítésére az egész életen át tartó tanulás során, és nem szabad megfeledkezni a szociális kirekesztéstől veszélyeztetettek sajátos szükségleteiről sem. Ennek segítségével fokozódhat a munkaerőpiaci részvétel és a gazdasági növekedés, miközben a társadalmi kohézió az oktatásba és képzésbe való beruházásnak megvan az ára, de az egyén, a gazdaság és a társadalom szintjén jelentkező busás nyereség révén közép- és hosszú távon megtérülnek a költségek.

TÁVOKTATÁS ÉS E-LEARNING

Az előző fejezetben láthattuk, hogy milyen fontos az élethosszig tartó tanulás (Long life learning – LLL), milyen módon kapcsolódik a világunk fejlődéséhez, fokozatos térnyerése minek köszönhető. Az atipikus tanulás ezen viselkedésforma alappillére, melynek két fő formája a távoktatás és az e-learning, azaz elektronikus tanulás. Elsőként tekintsük át, hogy e két fogalom konkrétan mit takar, milyen viszonyban állnak egymással.

TÁVOKTATÁS

Csoma Gyula 1987-ben a Felnőttképzési Kislexikonban az alábbi, ma is érvényes definíciót adta a távoktatásra:

„A távoktatás a távirányítás egyik sajátos módja; a távirányításra épülő irányítási, tanulási rendszer, amely a munka melletti tanulás körülményei között meghatározott, előírt és pontosan felépített ismeretek, gondolkodási – és korlátozottan – cselekvési műveletek elsajátítására szerveződik, meghatározott követelmények teljesítése érdekében. A tanulás feszes távirányítása arra törekszik, hogy a tanulás folyamata minden mozzanatát a kezében tartsa, ideértve a jártasságok, készségek kifejlesztését is, és a visszacsatolás-értékelés rendszerével megteremtse az önellenőrzést és ellenőrzést – ezzel pedig a tervszerű, eredményes továbbhaladás – optimális feltételeit. A távoktatás didaktikai rendszerében a tanítási-tanulási folyamat két szakasza térben és időben a lehető legtávolabb kerül egymástól. A közvetlen irányítás a lehető legszűkebb térre vonul vissza, a közvetett irányítás pedig a lehető legnagyobb mértékben kiszélesedik. Döntő fontosságúvá válik a tanulói önirányítás, amelyet azonban a közvetett irányítás rendszere nemcsak inspirál, hanem szabályoz is. A távoktatás a tanulók önfejlesztő energiáinak maximális felhasználására épül, mégis arra törekszik, hogy a tanulást (a tanulói önirányítást) közvetetten, de minél határozottabban befolyásolja és vezérelje. A tanítók és a tanulók közvetlen kapcsolatai megritkulnak, esetleg el is tűnnek. A közvetett irányítás a tanulóknál közvetlenül jelenlévő ismerethordozók és a tanulás irányítását megvalósító apparátus (metodikai instrumentumok) segítségével történik. Az ismerethordozók és a metodikai apparátus sorába a rádió, a televízió, a magnetofon, a telefon és a kábeltelevízió is bekapcsolódhat, jelenlétük (ill. egyikük, másikkuk jelenléte) azonban nem szükséges feltétele a távoktatás működésének, nem tartozik a rendszer didaktikai lényegéhez, viszont növeli

hatékonyságát. A tanulói feladatok központi kiértékelése komputerizálható, lehetnek olyan diszciplínák, ahol a tanulás irányítása vagy annak több szakasza komputerre kapcsolható. Ezek a megoldások is járulékos szerepet töltenek be.

A metodikai apparátus a tanulás ütemezését is előírja és szabályozza. Ebben fontos szerep jut a folyamatos visszacsatolás előre és pontosan meghatározott ütemezésének. A folyamatos visszacsatolás elsődlegesen nem önellenőrzés (de az is), hanem tanítói ellenőrzés és értékelés, amely távkapcsolat útján bonyolódik le, és legfőbb funkciója annak eldöntése, hogy az értékelt tanulási szakasz kellően eredményes-e ahhoz, hogy a következő elégséges alapjául szolgáljon.

A távoktatás didaktikai rendszerének az egyik leglényegesebb sajátossága az, hogy a tanulók csak akkor kapnak engedélyt a következő szakasz megkezdésére, ha a megelőző szakasz eredményei ezt biztosan lehetővé teszik. A távoktatás a tananyagot és a tanulási folyamatot szakaszokra bontja, minden szakasz végén visszacsatolási-ellenőrzési pontot iktat be.”

Ez a definíció azonban a technológia fejlődésének és az azóta bekövetkezett társadalmi változások miatt korrekcióra szorul. A számítógépes hálózatok, ezek közül is az Internet térhódításával az időbeli távolság részben eltűnik. A tanár-diák kapcsolat továbbra is szükséges, csak némileg megváltozik a szerepe.

A távoktatás a közhiedelemmel ellentétben nem azt jelenti, hogy a tanár és a tanuló közt nagy a távolság, esetleg telekommunikációs eszközön (pl. műholddal közvetített TV adás) át folyik az oktatás. A távoktatás alapvető jellemzője, hogy a tanárt a speciális módon megírt tankönyv vagy oktató szoftver helyettesíti. A tanuló a nap és a hét bármely szakában tanulhat, és ha problémája van, konzultálhat a tutorral, azaz a segítő tanárral. Ő azonban nem tanít, csak rávezet a megoldásra, ösztönöz, biztat, nagy ritkán helyesbíti a tévedéseket. S akármilyen meglepő, a távoktatásban a tanár – a könyv vagy software – közelebb van a diákhoz és alkalmazkodóbb, mint bármely más oktatási formában. A távoktatás speciális kialakítása azt jelenti, hogy a szerző(k) a célközönség ismeretében, az általuk befogadni képes stílusban és módon ismertetik a tanárt helyettesítő anyagban az új ismeretanyagot. A tanári ellenőrzést elvégzendő gyakorlatok helyettesítik, sokszor ezzel is bővítve az ismeretek átadásának módjait, mert a gyakorlatok önellenőrzése egyben a tananyag végiggondolását teszik szükségessé. Természetesen – s erről szeretünk elfeledkezni – ehhez a tanuló magas fokú önállósága és „tanulni tudása” szükségeltetik!

A távoktatás, mint módszer minden olyan ismeretanyag átadására alkalmas, mely nem igényel műhelyben történő közvetlen és hosszú időtartamú gyakorlatot. Amerikában pl. sebészorvos továbbképzésre is használják, míg teljesen természetesen a vegyészképzéstől a nyelvi oktatásig a közgazdász diploma megszerzésén át mindenütt a legszélesebb skálán mozognak a kifejlesztett tananyagok. Van, ahol felsőfokú képzésre használják leginkább, de szép számmal ismerünk középfokú képzéseket is a gazdaképzőtől a vállalkozási tanfolyamig, a nyelvoktatástól a számítógépes oktatásig. Nagyvállalatok saját munkatársaik egységes továbbképzésére külön tanfolyamokat fejlesztenek ki az erre szakosodott cégekkel (pl. banki ügyintézők, biztosítási ügynökök képzése, stb.), és tudunk olyan példát is, hogy sikeresen használják a távoktatást elektroműszerészeknek az új technikával való megismertetésére csakúgy, mint cégvezetők folyamatos továbbképzésére az új gazdasági szabályozók naprakész ismeretében. A távoktatás tehát eszköz, mellyel minden adekvát tartalom továbbítható. Az internethez hasonlóan legtöbb információ átadására alkalmas, kivéve néhány specifikus területet, ahol pl. a szaglás vagy ízérzékelés nélkülözhetetlen. (Ámbár borászati ill. borszakértői távoktatást is kifejlesztettek éppen Magyarországon.)

A távoktatás megjelenését 1840-re teszik. Ebben az évben kezdte el oktatni a gyorsírást Isaac Pitman levelezőlapokon Nagy-Britanniában, valamint ekkor vezették be a bélyeg alkalmazását is. Ennek köszönhetően a kezdetektől biztosítva volt egy kommunikációs közeg az oktató és az oktatott között, ami áthidalta a földrajzi távolságokat. Itt vált el egymástól térben és időben az oktatás és a tanulás folyamata. Az első eredeti levelező iskolát Berlinben 1856-ban alapította C. Toussaint, aki elsősorban nyelveket oktatott. Ugyanebben az évben vizsgáztak először olyan hallgatók a Londoni Egyetemen, akiket nem helyben oktattak. Itt választották külön az oktatást és a vizsgáztatást először. Ezután a világban sokfelé alakultak levelező iskolák: Franciaországban (1877), Amerikában (1891), Svédországban (1898). A következő fontos állomás az első közoktatási levelező iskolák megjelenése: Amerikában (1905), Ausztráliában (1914), Kanadában (1919), Új-Zélandon (1922).

A fejlődés következő szakaszát a tömegkommunikációs eszközök megjelenése idézte elő: rádió, televízió.

A BBC rádión keresztül sugárzott oktató műsorokat 1927-ben, ezt követően Amerikában is megjelentek a hasonló jellegű műsorok. Franciaországban 1937-től sugároz a rádió rendszeresen távoktatási műsorokat.

1939-ben megjelenik egy új médium a távoktatás palettáján és ez a telefon. Ezt a módszert Amerikában használják először mozgássérülteknél és otthon fekvő betegeknél. Később megjelenik a rádió-telefon is, 1942-ben Ausztráliában alkalmazzák először. Az első igazán

jelentős állami intézmény Franciaországban alakul 1939-ben (a mai CNED, Centre National d'Enseignement à Distance, az Országos Távoktatási Központ őse), ez az első nemzeti méretekben megvalósult program. A következő évtizedekben megjelenik a televízió is a távoktatás eszközei között, létrejön a levelező tagozatos képzés. A kezdeti alapszintű képzések mellett megjelennek az egyetemi szintű képzések is.

A 80-as évektől úgy lehet tekinteni, hogy a távoktatás mindenütt létezik a világon, bár a megvalósult modellek igen sokfélék. A 80-as és 90-es években terjed el a számítógéppel támogatott oktatás. A 90-es évek elejétől hatalmas fejlődésnek induló világméretű számítógépes hálózat, az Internet elkövetkező hatását a távoktatásra ma még fel sem lehet becsülni.

Európa fejlett országaiban egyre inkább tért hódítanak az atipikus munkavállalási formák. S bár nálunk e foglalkoztatási formák képtelenek voltak nagyobb tömegeket megérinteni, azonban a foglalkoztatás bővülésének ilyen irányú nemzetközi tendenciái, egyre inkább érvényesülnek. Az európai integráció folyamatában a magyar munkaerő minőségének javítása számottevően erősítheti pozíciónkat, és már rövidtávon számottevően hozzájárulhat a gazdaságban létrehozott hozzáadott érték növekedéséhez. A nemzeti és regionális fejlesztési politikák világszerte a versenyképességet döntő módon befolyásoló forrásokként kezelik az emberi erőforrásokat. E törekvés egyik jó példája az Európai Uniónak az a célkitűzése, amely a *tudás Európája* megvalósítását állítja középpontba. Ez közvetlenül kapcsolódik a teljes életpályára kiterjedő tanulás gondolatához.

A távoktatás, távtanulás a hagyományos képzéstől sok tekintetben különbözik. Olyan komplex rendszer, amely a hallgatók, és a piac szükségleteire alapozva irányított tanulási folyamatot eredményez. A követelmények teljesítése érdekében a tananyag feldolgozását meghatározott eszközökkel vezérli.

A Nemzetközi Távoktatási Tanács adatai szerint ma a világon több mint 11 millió ember vesz részt távoktatásos képzésben. A távoktatási formában működő intézmények átfogják a világot, lehetőséget adva a változó igények kielégítésének, kihasználva az oktatásban a műszaki fejlődés által teremtett új lehetőségeket, amelyek a mainál lényegesen nagyobb népesség bekapcsolódását biztosíthatják a hatékony szakképzésbe.

A nyitott szakképzés iránt megnyilvánuló igény a távtanulás fogalmát is új megvilágításba helyezi. A munkaerőpiacon már jelenlévők szakmai mobilitásának javítása, a hatékonyabb, esetenként a földrajzi távolságokat is kezelni képes munkaszervezési módként a távmunka

jóval sürgetőbben igényli a távtanulás lehetőségeit, mint a lassan s az intézményi tehetetlenség miatt éppen ellenérdekeltségével is küszködő hagyományos oktatási rendszer.

A távoktatás, távtanulás lényegesebb jegyei:

- tanulás - oktatás időbeni és térbeni elkülönülése,
- irányított tanulás,
- önállóság alapkövetelménye,
- kreativitás,
- a tanuló és a tanár kapcsolata,
- a hallgató és az intézmény sajátos kapcsolata,
- a kommunikációs eszközök jelentőségének növekedése,
- ellenőrzés, visszacsatolás,
- speciális tananyag és oktatócsomag.

A távoktatás felhasználási területeit bővíti a *nyitott szakképzés* lehetősége. A nyitott szakképzés lényege, hogy a képzésben részt vevők számára, időben és térben jelentős megkötöttségek nélkül az elsajátítandó tananyagot, ismereteket, valamint az elsajátításra vonatkozó útmutatásokat és a megszerzett tudás ellenőrzését, módszertanilag alaposan kidolgozott különböző információhordozók: nyomtatott anyagok, audiovizuális, számítógép információhordozók: CD és DVD, sőt, manapság a Blu-Ray korongok tartalmazzák. E képzés esetén lehetőség van a tananyagok hálózatokon (intranet és internet egyaránt) keresztüli elérésére is. Az elmúlt időszakban a nyitott oktatás körébe tartozó képzési és tanulási módszerek diverzifikációjának lehattünk tanúi, s az ide sorolt képzési módszerek az ismeretelsajátítás egyre szélesebb területeire terjednek ki.

Fontos tulajdonsága a nyitott képzésnek a hagyományos oktatási képzési elemek (tudományos és gyakorlati ismeretek és alkalmazásuk, pedagógiai értékek, kulturális tartalom és háttér) és az iparszerű tevékenység (igényfelmérés, erőforrás-allokáció, tervszerűség, gazdaságosság, csoportmunka, szervezetheg, minőség-ellenőrzés, marketing, menedzsment, hatásvizsgálat) kombinációja, kölcsönhatása. Ha sikerül a korszerű informatikai eszközökhöz való széles körű hozzáférést biztosítani, e módszerek jelentősen javítják a tanulási, és ezáltal a társadalmi esélyegyenlőséget is.

E-LEARNING

Az e-learning, azaz elektronikus tanulás az egyik legerőteljesebben fejlődő atipikus oktatási forma, a távoktatás egyik formája. Az Ageless Learner nevű e-learning oktatási portál [\[1\]](#) definíciója alapján az e-learning az online tanulás területeit leíró web-alapú valamint technológiák által támogatott képzési forma:

- az internet és egyéb hálózati funkciók és a tanulás konvergenciája, illetve az oktatás digitális támogatása,
- a tananyagok digitális előállítása, a tanulás támogatása, eljuttatása a célcsoportokhoz,
- a tanulók és tanárok (tutorok, szakemberek) összekapcsolása,
- a forma lehetőséget ad a folyamatos nyomonkövetésre, elszámolhatóságra,
- folyamatos hozzáférést biztosít a tananyaghoz, lehetőséget ad a résztvevőknek a változások gyors követésére.

Az e-learning tananyagok és infrastruktúra helyes kialakításával mindez megvalósítható. Sajnos, sokszor előfordul, hogy az emberek többsége inkább az „e”-t helyezi előtérbe a „learning”-gel szemben. Az e-learning hatékonysága nagyban fokozható a tananyag tanulóközpontú kialakításával. Alapértelmezésben is egyértelmű, hogy a jól tagolt (akár papíralapú) tananyagok a tanulók számára sokkal jobban tanulhatóak, s amennyiben jól értelmezhető ábrákat, táblázatokat, segítő részeket tartalmaznak.

SZEREPEK

Helyezzük egymás mellé a két tanulási formát. Látható, hogy mind a hagyományos, mind a távoktatásos formának megvan a maga helye és szerepe az oktatási rendszerünkben, csupán a helyén kell kezelni őket. A távoktatás egy új szerep. Tanulni és meg kell tanulnia a hallgatónak, hiszen új eszközökkel, más környezetben végzi a tanulás folyamatát. Eddig egy osztályban, több tanulótársával együtt, tanár útmutatásával végezte a feladatokat, most valószínűleg egyedül, tanár nélkül (de tutor segítségével) akár otthonról, bármilyen időpontban, időben és térben tehát függetlenül oldhatja meg a feladatokat, sajátíthatja el a tananyagot. A régi passzív, befogadó hagyományos diákszerep felcserélődik egy aktív, önálló, partneri szerepre. Nagy szerepet kap a tanulásra való motiváltság, a kommunikációs lehetőségek ismerete, hiszen a tanári motiváció egészen más szerepet ölt, s a közvetlen kommunikáció is átalakul. Az új kommunikációs stratégiák elsajátítása tehát a távoktatás nagyon fontos előfeltétele:

- **számítógépen keresztül történő kommunikáció:** segítheti a csoportos, együttműködő módszerek kialakítását, mint pl. páros, vagy csoportmunka végzését,
- **a web, mint információforrás:** a web segítségével felhasználhatók az interneten található információforrások (bár ezek minőségéről folyamatos viták zajlanak), de mindenképpen hasznos lehet, pl. adatkommunikáció céljára,

Az új módszerrel nem csak a hallgató hozzáállásán kell változtatni, de ugyanúgy a tanár szerepe is jelentősen megváltozik. A terminológia immár nem is tanárnak, hanem tutornak, facilitátornak, tanácsadónak, esetleg szupervizornak hívja. A nevekből is látható, hogy a szerepe milyen irányba változott: nem magyarázó, kézen fogva vezető, hanem inkább irányító, segítő, az útvonalat mutató személyé vált, vagy ahogy a „Mátrix – Forradalmak” című filmben is hallhatjuk: *„Én csak az ajtót mutatom meg, de neked kell belépned rajta!”*. Megváltozik a tutor felelőssége, máshogyan kell beosztania az idejét, máshogyan kell a bejövő információval gazdálkodni. Ki kell alakítani az új módszereket, amely segítségével meg tud felelni a szerepének. Ehhez új készségeket kell kialakítania:

- az on-line folyamatok megértése, a meglevő struktúrák és aktivitások hatékony használata,
- technikai készségek, mint számítógép-kezelés, internet hatékony használata,
- on-line kommunikációs képességek, mint az időgazdálkodás képessége, személyes interakciók on-line használatának képessége: humor, érzelmek, probléma- és konfliktuskezelés,
- szakmai jártasság, ami már nem csak a klasszikus értelemben vett szakmai tudást jelenti, de bekapcsolódnak a legfontosabb on-line források ismeretei, a legújabb információs, kapcsolattartási lehetőségek széles körű ismerete is: wikipedia, msn, skype, facebook, iwiw, myvip, mspace, youtube, és még sorolhatnám a lehetőségeket. Bizony a számítástechnikai jártassághoz ma már hozzátartoznak ezek a kommunikációs csatornák is.

Foglaljuk össze táblázatos formában a különbségeket, hogy hogyan alakulnak a szerepek az új rendszerben! Ehhez Kovács Ilma (1997) segítségét kérjük, aki így összegezte a fő sajátosságokat:

AZ OKTATÁS ÉS A TANANYAG

HAGYOMÁNYOS OKTATÁS	TÁVOKTATÁS
Passzív befogadás és megszabott ritmusú, mélységű oktatás.	Általánosan felmerülő új igények mind a hagyományos, mind a távoktatás vonatkozásában: Az oktatás individualizálódik: egyéni ritmus, mélység lehetősége. A tanuló aktív, önálló szerepű.
Zárt, kész tudásátadás, melynek forrása a tanár, tananyag. Zárt, lineáris tanulási környezet.	Nyitott, multi- és hipermediális tanulási környezet.
A tananyag kiegészítő, járulékos szerepű a tanítói munka mellett.	A tananyag az egyéni tanulást szolgálja. A tartalom mellett hordozza a tanítási módszereket is, ami sajátos tanulási módszert indukál. A tananyag funkciója: a tudás aktív megszerzésének elérése.

A TANULÓI SZEREP

HAGYOMÁNYOS OKTATÁS	TÁVOKTATÁS
Életkor és tudásszint szerint homogén csoportok.	Homogén csoport valószínűsége kicsi. (Felnőttek esetén egyéni tananyag-kezelési módszerek figyelembe vétele.)
Közös térben tanulnak.	Térben, esetleg időben távoli tanulás.
Állandó ellenőrzés, függő viszony a tanártól.	Közvetlen ellenőrzés nélküli, tanártól független munka.
A tanuló személyes elérhetősége miatt kevesebb nyilvántartás.	Gazdag információtár szükséges a tanulóról.
Nincs szükség szervezett nyomon követésre.	Szervezett, folyamatos nyomon követés az önálló, önismereten feltételező tanulás segítéséhez.

A TANÁRI SZEREP

HAGYOMÁNYOS OKTATÁS	TÁVOKTATÁS
Homogén tanulói csoportok oktatása a feladat: kor, tudásszint, szakmai képzettség szempontjából.	Heterogén csoportok valószínűsége: kor, munkahely, munkatapasztalat, részvételi motiváció szempontjából. Nagy létszám, ismeretlen tanulók.
Szemtől-szemben oktatás.	Soha nem vagy ritkán látja a tanulót a tanár. A kommunikáció előre gyártott, de "egénre szabható" tananyag, vagy telekommunikáció útján történik.
A tanár ismeretátadó.	A tanár közvetítő, támogató (mentor, tutor) az önálló tanulásban. Nem tanít, hanem a tudás megszerzését irányítja, facilitálja.
Központi szerepű a tanuló-tanító kapcsolat.	A dialógus lehetősége, fenntartása problematikus, korlátozott.

SZEREPPROBLÉMÁK

Idillikusnak tűnhetnek a fenti táblázatok. Mindenki tökéletes, mindenki tudja, mit kell tennie és el is végzi azt. Ez persze csak a legritkább esetben történik így. Milyen problémák merülhetnek föl a távoktatásos képzési formák alatt?

Ha végigmegyünk a táblázatokon, könnyű meghatározni a lehetséges gondokat: nincs megfelelő tananyag, megfelelő tanítási módszer; a hallgató nem eléggé motivált, a tanár nem kellően mentorál. Ezek egyértelmű problémák lehetnek, amelyek ellen könnyű (bizonyos fokig) tenni. Viszont a mélységekben ennél bonyolultabb a helyzet. Tapasztalataim szerint az egyik legfontosabb gátló tényező a *szorongás*. Igen, talán meglepő lehet, hogy pszichológiai alapra tévedtünk, de az előző 50-60 képzésem során (ebből csupán 2 képzés volt saját vezetésű, távoktatásos jellegű) ezt a problémát volt a legnehezebb leküzdeni, sokszor sikerült, de bizony, sokszor – minden erőfeszitésem ellenére – nem bizonyultak a módszereim sikeresnek.

Mitől szorong a hallgató? Az első szorongása (s ez általános) az új helyzetből adódik. Amennyiben ez hagyományos oktatás, akkor az új csoport, a csoporttársak, az új tanár, az előttünk álló feladatok, vizsga, mind-mind szorongásra okot adó tényezők. Mivel a munkahelyemen általában munkanélküliek átképzésével foglalkozom, nálunk még ehhez hozzákapcsolódnak a munkahely hiányából fakadó szorongás, illetve egyéb szociális problémák (anyagi, családi). De mindezek a szorongások egy akármilyen képzésnél megjelennek. A távoktatás esetében ezek egy része eltűnik, ám jönnek helyette mások:

időbeosztás (határidők, más feladatokkal való összeegyeztetés), technikai problémák (internet, számítógép elégségesége, működőképessége). Talán a legfontosabbat hagytam a végére: a „komputer-szorongás”. Egy 2004-es felmérés alapján a felsőfokú oktatásban résztvevő hallgatók 50%-a érez bárminemű szorongást az informatikával kapcsolatban. Ha ez a szám a felsőfokú oktatásra vonatkozik, vajon mekkora lehet ez az érték a saját területemen, ahol 8 általánostól kezdődnek a végzettségek, felsőfokú végzettség csak igen ritkán jelenik meg. Saját tapasztalatom alapján ez bizony eléri a 80-90%-ot is. Egyes csoportokban csupán 1-2 fő az, aki hozzá mer nyúlni a képzés elején a gépekhez. Természetesen ez az iskolarendszerű képzés alatt elsajátított informatikai tudásszint növekedésével változik: aki kijön az általános iskolából, az már rendelkezik egy bizonyos szintű informatikai írástudással. Ez viszont nem jelenti azt, hogy a szorongása a tudásszint növekedésével elmúlik...

Ez a szorongás még veszélyesebbé válik a távoktatásos forma esetében. Ahhoz, hogy a félelem oldódjon, a hallgatónak igénybe kell vennie az informatikai kommunikációs formákat, hogy a tutor megfelelő módon tudja kezelni a problémát. De pont ettől fél a hallgató! Fél, hogy kért okozhat a számítógépben, az egoját érzi fenyegetve, például attól, hogy nevetségessé válik, de érezhet kontrollvesztést is a számítógéppel szemben. Azt már meg sem említem, hogy sokan a túlságosan bonyolult gépekkel szemben alapvetően megtartanak bizonyos távolságot. „Ehhez én úgysem értek, ha hozzányúlok, biztosan elrontom!” Ha azt szeretnénk, hogy hallgatóink sikeresen elvégezzék a kurzust, legalább két dolgot ki kell alakítanunk bennük:

- számítógépes magabiztosság (computer confidence), ami azt takarja, hogy a az egyén úgy érzi, képes megtanulni a számítógép kezelését, s képes azt használni,
- számítógép kedvelés (computer liking), ami azt fejezi ki, hogy az egyén mennyire élvezi a számítógéppel végzett munkát.

A számítógépes magabiztosság minden távoktatásos képzés alapja. Az általam lebonyolított mindkét képzésnek előfeltétele volt az ECDL-szintű informatikai írástudás. Ez természetesen nem minden esetben megvalósítható, hiszen nem csak informatikusok fognak távoktatásban részt venni. Emiatt kidolgoztunk egy 12 órás „Informatikai felkészítés” nevű tárgyat, amely a hagyományos módszerekkel próbálta kialakítani a computer confidence élményt. Természetesen csupán egy redukált ismeretanyagot tudtunk átadni e rövid idő alatt: az operációs rendszer alapjait, gépkezelést, s némi internetes tudásanyagot. Csak annyit, hogy maga az oktatás zökkenőmentesen zajlódjon le, de legalább annyit, hogy a hallgató fel tudja venni a kapcsolatot a ttorral, s válaszni tudja a fennálló problémát részére.

A számítógép kedvelés az előzőre épít. Ha már tudom használni a számítógépet, megpróbálom a saját céljaimra használni azt. Meg kell találjam azokat az informatikai

lehetőségeket, amelyek engem érdekelnek, s meg kell tanulnom minél hatékonyabban kihasználni a számítógépet ezen cél érdekében. Fel kell ismernie a hallgatónak a számítógép hasznosságát és a használat könnyűségét, meg kell tanulnia használni az új médiát céljai eléréséhez. Csak apró megjegyzés az új médiával kapcsolatban: meglepő, hogy még a több éve számítógéppel dolgozók is előnyben részesítik a nyomtatott tananyagot, mint a digitálisat, függetlenül attól, hogy ők már tisztában vannak a digitális technológia előnyeivel. Inkább elveszítik a tananyag interaktivitását, de papírról tanulnak szívesebben, mint monitorról.

Eddig csak arról beszéltem, hogy a hallgató milyen problémákkal találkozhatja szembe magát. De a játék legalább kétszereplős, ráadásul van egy passzív résztvevője is: a tutor és a tananyag. Ha bemennék egy középiskolába, s megkérdezném a tanárokat (most hagyjuk el az informatikát), hogy hányan készítenének szívesen digitális tananyagot, s hányan szeretnének tutorként levezetni egy-egy képzést, vajon milyen százalékban kapnék pozitív visszajelzéseket? Bizony, a tanár is ember. Ők is ugyanúgy táplálhatnak ellenérzéseket az informatikával szemben. Hiszen dolgozni kell vele, új tanítási formát elsajátítani (miért is nem jó a régi, bevált módszer?), ráadásul számítógéppel kell fenntartani a kapcsolatot a hallgatóval... Amíg a tanár nincs meggyőzve (meggyőződve) arról, hogy bizonyos helyzetekben ez jó módszer, hogy a hallgatóknak előnye származhat ezen atipikus módszerek használatából, hogy néha csak ez marad, mint lehetőség, addig csupán erőltetésnek tűnik minden kezdeményezés. Egyik egyetemi tanárom osztotta két csoportba az embereket informatikai szempontból: bennszülöttek és bevándorlók. Találó a két kifejezés: a bennszülöttek azok, akiknek a számítógép immár természetes. El sem tudják képzelni az életüket nélküle, olyan része az életüknek, mint a mobiltelefon, vagy a televízió. A bevándorlók pedig, akik csak „felvették” a számítógépes attitűdöt, csupán a legszükségesebb esetben fordulnak a géphez, s lehetőleg elkerülik, kikerülik azt. Olyan lehet ez, mint a mobiltelefon nagyszüleink számára. A legtöbb felnőtt ember a második csoportba tartozik, még csak most kezd kialakulni az a társadalmi réteg, akik ténylegesen bennszülöttei az informatikai társadalomnak. Ők a gyermekeink. Amilyen gyorsan fejlődik a számítástechnika, ezek a sorok már valószínűleg 10-15 év múlva értelmüket veszítik. De addig is térjünk vissza az eredeti fonalhoz.

Ha a tanár igényt érez arra, hogy tutorként működjön közre az oktatásban, már csak egy jó tananyag szükséges, amely valamilyen szinten helyettesíti a tanári iránymutatást. Beszéljünk egy kicsit erről is!

A *hagyományos tananyag* általában tankönyvből és munkafüzetből áll. A tankönyv vonatkozó részeinek feldolgozása után a tanuló elvégzi a kijelölt feladatokat a munkafüzetben, a tanár segítségével ellenőrzik azt, kijavítják a hibákat, elvégzik a visszacsatolásokat, s ezzel a kör bezárult. A távoktatás is hasonló elven működik, csak ahogy láttuk, a szerepek picit megváltoztak. Elsődleges, hogy a távoktatás nem mindig jelenti digitális tananyag meglétét. Az alapot adó, ismeretközlő forrás lehet egy egyszerű, hagyományos tankönyv is. A feldolgozás módjában van a különbség. A hagyományos rendszerben a tanár segítségével, lineárisan dolgozzuk fel a tananyagot, elsősorban a tanár magyarázatai szolgálnak a bevésésre. Itt megfelelő instrukciókat kap a hallgató digitális formában, amely meghatározza a feldolgozandó tananyagegység forrását („Olvasd el és értelmezd a tankönyv 3. fejezetét, emeld ki a kulcsszavakat!”), de nem utolsósorban meghatározza a feldolgozáshoz szükséges tevékenységi formákat, attitűdöket és nevelési elveket. („A feldolgozás során gondolkodj el azon, hogy a saját háztartásodban milyen módon tudnád az energiafogyasztást csökkenteni. Készíts táblázatot a legtöbbet fogyasztó eszközökről, s készíts heti naplót, mennyi ideig használsz őket!”) Látható, hogy a meghatározott feladat inkább a feldolgozás mélységét növeli, nem feltétlenül a kompetenciákat erősíti.

Természetesen a tananyag megjelenhet *digitális formában* is. Egyik legelterjedtebb formája az *on-line tananyag*. Az internetre feltöltött kurzusok egy jól használható keretrendszert kapnak (pl. Moodle, aTutor), amelyek az oktatás nélkülözhetetlen adminisztratív feladatait is elősegítik, megoldhatóak a kommunikációs feladatok, s a hallgatók folyamatos nyomon követése is biztosítva van. Folyamatosan láthatjuk az aktivitásokat, a rendszer tanárként is funkcionál: instruál, javít, értékeli. Hátránya, hogy a tanulás folyamán folyamatos internet kapcsolatra van szükség, a hallgató nem tudja letölteni magával vinni a tananyagot.

Ha lemondunk a keretrendszer hasznos oldaláról s átvállaljuk az adminisztrációs feladatokat, használhatunk *elektronikus tananyagot* is. Ebben az esetben egy adathordozón (CD, DVD, pendrive, hálózati mappa, internetes megosztás) biztosítjuk a hallgató számára a feldolgozandó tananyagegységeket, ami jó esetben valamilyen szintű interaktivitást is biztosít, pl. feladatokat és azok kiértékeléseit is tartalmazza. Ezek az anyagok feltelepíthetőek egy számítógépre, laptop esetén bárhol böngészhetőek, tehát tényleg megvalósítható az időbeli és térbeli függetlenség a tanulásban. Hátránya az on-line tananyaggal szemben a statikussága: a már kiadott tananyagot igen nehéz frissíteni, javítani, továbbá hogy nincs nyomon követés, legalábbis a hagyományos értelemben. Meg lehet itt is oldani a hallgatók

kontrollját, hiszen a feladatok tartalmazhatnak „mérőköveket”, ahol valamilyen produktummal kell a hallgatónak jelentkezni a tutornál, pl. beküldeni egy elkészült anyagot, vagy bizonyos időszakonként egy-egy záró feladatsort elvégezni on-line módon, számot adva az elvégzett tanulási folyamatról.

A következőkben az elektronikus tananyagot vesszük jobban górcső alá. On-line tananyagot már sokan készítettek, sőt, egyre inkább ez az iránymutató. A keretrendszerek sok terhet levesznek a tutor válláról, de valahol korlátozzák is a hallgatót. Lássunk egy olyan példát, amely egy pendrive-ra mentve is megállja a helyét, ismerjük meg a készítés folyamatát, a folyamat során azokat az elemeket, amelyek szükségesek a hatékony munkavégzéshez. Nézzük meg, hogy a célcsoportok hogyan befolyásolják a tananyagot, milyen szempontokat kell figyelembe vennünk a kidolgozás folyamán. Hosszú munka előtt állunk, de kezdjük az egészet egy kis elmélettel!

HOGYAN KEZDJÜNK HOZZÁ?

Mint említettem, a feladatunk az lesz, hogy olyan digitális tananyagot állítsunk össze, amely bárhol futtatható lesz. A bárhol bizony elég tág fogalom: nem korlátozhatjuk egy adott operációs rendszerre, vagy kizárólagosan internet használatra. Tehát olyan tananyagot kell létrehozzunk, amely *cross-platform*-os, azaz bármilyen operációs rendszer és hardver alatt működőképes, továbbá biztosítsa úgy az interaktivitást, hogy ne kelljen internetre csatlakozni. Eléggé bekorlátoztuk magunkat, nem sok lehetőségünk maradt. Egyik ilyen rendszer, a HTML. A HTML fájlok minden rendszeren megjeleníthetők, akár személyi számítógépről, akár mobiltelefonról (okostelefonról), PDA-ról van szó. Nem szükséges a megjelenítéshez internet hozzáférés sem, csupán egy böngésző, amely képes értelmezni a HTML parancsokat, tag-eket. Amit használhatunk mellette, hogy kicsit dinamikusabbá tegyük az oldalunkat, az a CSS, a Javascript és a Flash. Azaz tananyagként interaktív weboldalt fogunk gyártani. Sajnos, PHP-t, vagy egyéb szerver oldali nyelvet nem alkalmazhatunk, pedig igen megkönnyítené a feladatunkat. Viszont ahhoz már egy webszerver szükségeltetik, azaz nem árt az internet kapcsolat. Akkor a hozzávalók:

- HTML (a tartalomhoz)
- CSS (a formázáshoz)
- Javascript (dinamikus részekhez, pl. feladatokhoz)
- Flash (bemutatókhoz, feladatokhoz)

Ha ezt a négy hozzávalót birtokoljuk (nem csak fizikailag, de tudásilag is), akkor hozzákezdhetünk az első digitális tananyagunk létrehozásához. De pillanat, ezek csak a nyelvek, amelyen elmondhatjuk, hogy mit akarunk! Szükség lesz még néhány eszközre is:

- forráskódszerkesztő (notepad, total commander, far manager, notepad++)
- Flash Editor (Adobe Flash)
- valamilyen képernyőörögzítő applikáció

Amit még nem említettem: türelem, kitartás, akaraterő. Azt mondják, hogy egy tanítási óra „e-learning-esítése” kb. 6-10 órai munkát jelent. Természetesen, ha már létezik maga a tananyag valamilyen analóg formában (lehet ez akár a fejünk is ☺), vagy akadnak feladataink a témakörhöz, a nagymértékben csökkentheti ezt az időt. Ettől függetlenül nem lesz kevés a beleölt munka, de csak egyszer kell megcsinálni, utána már csak használni kell.

Tartsanak velem!

TÉMAVÁLASZTÁS, CÉLCSOPORT

Mivel informatikai jellegű maga a szakdolgozat is, innen jön a téma is: adjuk át webes ismereteinket a diákjainknak. Természetesen, ez egy elég széles témakör. Korlátozzuk a HTML ismeretekre, azaz tanítsuk meg a hallgatót a legfontosabb HTML parancsokra, a formázásra, linkek, képek beszúrására, táblázatok és űrlapok kezelésére. Ez sem kevés, de legalább egy teljes képet tudunk hallgatóinknak nyújtani a webszerkesztés alapjairól.

Kiknek szól ez a tananyag? A legfontosabb kérdések egyike. Milyen végzettséggel rendelkeznek hallgatóink? Milyen előzetes tudásuk van? Teljesen más a kép, ha általános iskolásoknak készül, ha időseknek, ha érettségizetteknek, ha már legalább középfokú ismerettel kell rendelkezzen a hallgató. Mivel a webtervezés az OKJ-ban is megjelenik a Webmester szakképzésen belül, fogadjuk el ennek a szintjeit: érettségizett, középfokú informatikai ismeretekkel rendelkező hallgató lesz az alanyunk. Ez olyan szempontból kedvező, hogy nem kell elmagyarázni neki a számítógépkezelés rejtelseit, bár ez a modul nem is arról szeretne szólni. Viszont bátran használhatunk szakkifejezéseket, építhetünk az előzetes tudásukra. Nem szabad elfelejteni azt sem, hogy legtöbbször azért nem szabad elfogadni azt, ha a hallgató „bemondás” alapján nyilatkozik a tudásszintjéről. Az sem mindig célravezető, ha papírt kérünk tőle. Pontosan ezért nem azt fogalmaztam meg, hogy pl. „Számítástechnikai szoftverüzemeltető” végzettsége van. Készítsünk egy tesztet, amellyel felmérhetjük a témához kapcsolódó előzetes kompetenciákat. Ezt hívjuk PLA-nak (Prior Learning Assessment), azaz előzetes tudásszint felmérésnek. Mivel még nem rendelkeznek a hallgatók a téma kompetenciáival (nem tudnak weblapot szerkeszteni), nem kérdezhetünk tőlük olyat, ami majd később, a képzés alatt fog megjelenni. De kérdezhetünk tőlük olyat, amely ahhoz szükséges, hogy a majdani kompetenciákat el tudják sajátítani. Itt például kérdezhetünk tőlük fájlkezeléssel kapcsolatos tudáselemeket: másolás, törlés, átnevezés, tömörítés, keresés. Vagy bizonyos grafikai ismeretekről is meggyőződhetünk: pixel- és vektorgrafikus objektumok különbségei, konverziók. Összefoglalva olyan ismereteket kérdezzünk meg, amely nélkül nehéz lenne a valódi tananyagot elsajátítani.

ELŐMUNKÁLATOK

Ha már tudjuk, kinek és mit szeretnénk oktatni, pontosítsuk a témát. Rendben, hogy HTML oktatás, de bontsuk ki egy kicsit. Konkretizáljuk a feladatokat, a részeket, bontsuk fel az egészet rész célkitűzésekre. Mondjuk meg, hogy melyik fejezetben mit szeretnénk megtanítani. Erre szolgál a *forгатókönyv*. Nem kell megijedni, nem fogjuk a rendezők és forгатókönyvírók kenyerét elvenni. Készítsünk egy egyszerű táblázatot, amelyben leírjuk, mit szeretnénk oktatni!

1. Hogyan tanuljak?

2. A HTML alapjai

- a. Előkészületek
- b. A HTML kliens/szerver architektúrája
- c. A HTML, mint lapleíró nyelv
- d. HTML kód írása
- e. HTML kód tesztelése

3. Formázási műveletek

- a. A HTML oldal struktúrája
- b. A HTML tag-ek felépítése
- c. Lapformázás HTML-ben
- d. Bekezdésformázás
- e. Karakterformázás
- f. HTML editorok
- g. Gyakorló feladat

4. Linkek, képek

- a. Képek beszúrása
- b. Hivatkozások készítése

5. Táblázatok

- a. Táblázatok készítése
- b. Táblázatok formázása
- c. Gyakorló feladat

6. Űrlapok

- a. Űrlapok készítése
- b. Szövegbeviteli mezők
- c. Kapcsolómezők - jelölőnégyzet
- d. Kapcsolómezők - rádiógomb
- e. Kapcsolómezők - lista
- f. Nyomógombok
- g. Űrlapok kezelése Javascript segítségével

h. Gyakorló feladat

7. HTML alternatívák

a. Forráskód szerkesztők

b. Dreamweaver

8. Önellenőrzés

9. Feladatbank

Természetesen, egy forgatókönyv mindig attól függ, hogy ki és kinek készíti el. Azonban van néhány elem, amit érdemes szerepeltetni. Ilyen a „Hogyan tanuljak?” fejezet. Mivel – ahogy már említettem – egy egészen új tanulási formáról van szó, érdemes pár mondatban leírni, hogyan használja a hallgató a tananyagot. Nem kell oldalakat szánni a leírására, de az irányelveket tisztázni kell. Szintén érdemes egy Feladatbankot is létrehozni. Ez tartalmazza a különböző fejezetek gyakorló feladatait, lehetőleg megoldással együtt. Ha a hallgató minden feladatot becsülettel elkészíti, biztosan megtanulta a tananyagot, elsajátította a kompetenciákat.

HOGYAN TANULJAK?

Fejtsük ki egy picit részletesebben, miért van szükség erre a részre! A hagyományos ismeretelsajátítás a tanár segítségével történik, voltaképpen az egyetlen információszerzési csatorna. A távoktatás esetében ez esik ki. Mi marad helyette? Írtam már a különböző forráslehetőségekről, ezt az információt a hallgató számára is elérhetővé és ismertté kell tenni. Én egy ábrával szoktam szemléltetni az információszerzés forrásait és sorrendjét:



Láthatjuk, hogy a fenti négyes az önálló tanulást erősíti. Csak akkor szabad a tutorhoz (instruktorhoz) fordulni, ha már minden egyéb lehetőséget kimerített a hallgató. Az első ajánlott lépés a tananyag újbóli átnézése. Mivel probléma esetén a hallgató magára a kérdésre koncentrál, új szemszögből fogja átnézni a tananyagot, könnyebben emeli ki az összefüggéseket. Az önálló, pozitív tapasztalás ráadásul sikerélményt hordoz magában, nem beszélve arról, hogy mélyebb bevéséssel jár.

Természetesen megvan az esélye, hogy nem kap választ a kérdésére. A megfelelő kommunikációs csatornák használata pozitív hatással van a hallgatók hatékonyabb, egymás közötti munkájára, s nem utolsósorban szocializációs hatásai is vannak. Érdekes módon, az IRL (in real life, azaz a valós életben) jellegű attitűdök lemásolódnak a virtuális közösségek kapcsolataira is. Ha valaki visszahúzódó a valós életben, a chat, illetve a fórum csatornákon sem töri magát. Ami nagyon hasznos, hogy szinte minden kurzusban megjelennek az oktatói vénák, azaz az olyan egyedek, akik szívesen segítenek a többieknek, sőt, szinte várják a többiek problémáit, hogy megoldhassák őket.

A harmadik szint a virtuális kommunikációs csatornák használata. Ez elsősorban a keresőoldalakat jelenti. Bár legtöbbször viszonylagos biztonsággal kezelik a google, bing, yahoo, stb. keresőket, egy bonyolultabb keresés már kifoghat rajtuk. Nem egyszerű dolog megfogalmazni a problémát úgy, hogy azt egy kereső is megértse, hogy értelmes, használható találatokat adjon ki. A keresők által adott találatokból kiszűrni a saját problémánkra megoldást adó találatokat, majd abból kiemelni a választ... No, ez már az önálló tanulás egy fejlett módja!

Negyedik szintje az információszerzésnek a tutor megkérdezése. Természetesen egy távoktatási rendszerben sem folyamatosan elérhető az instruktorként. Vannak konzultációs időszakok, amikor a hallgató felvetheti a kérdéseit, illetve a tutor megválaszolja a beérkezett problémákat. (A konzultációnak természetesen lehet személyes és virtuális módja is, az első a tutor és a hallgatók személyes részvételével történik, az utóbbi pedig a virtuális térben: chatcsatornákon, fórumokon, email formában működhet.) Tényleg ez legyen az utolsó mentsvár! Ha már elkezdjük a távoktatásos formát, ne menjünk vissza a hagyományos formába, ha nem muszáj!

CÉLKITŰZÉS

A fenti forgatókönyvben 6 olyan fejezet található, amely a szakmai kompetenciákat dolgozza fel. Ha azt szeretnénk, hogy a hallgató motiválva és céltudatosan induljon neki egy-egy fejezetnek, tudassuk vele a céljainkat, amelyek az ő céljai is lesznek.



A tanulási egység célkitűzései

Elérkeztünk egy olyan részhez, amely segítségével sokkal szívesebb, esztétikusabb weblapot leszünk képesek készíteni. A lapok végén megint megjelennek a gondolkodtató feladatok, nézgesd őket, gyakorlati problémákat vetnek fel, amelyek később bármikor előjöhetnek a saját munkádban is. Amennyiben kérdésed merülne föl a megoldások közben, próbálj visszatérni a megfelelő lapra, ha pedig ott sem találtad meg amit kerestél, vagy nem egyértelmű a kérdés, tudod: fordulj bizalommal konzulensedhez.

Webmester E-learning



E részecélkitűzés megtanulása után az alábbi tudáselemekkel leszel gazdagabb:

- 1) Megismered a linkek beszúrásának formáit.
- 2) Láthatod majd, hogyan kell képeket beilleszteni, méretezni.
- 3) Megtudod, hogyan kezelheted a képek és az adattartalmak kapcsolatát.

Bár rövid a rész, annál fontosabb ismeretanyagot tartalmaz, tehát próbáld minél hatékonyabban kihasználni a rendelkezésre álló időt.

Itt az ideje némi kreativitásnak! Jó munkát!



A pillanat szüleménye: A merevlemez feje megrongálódott. Kérem, rakja be a sajátját!

Fontos, hogy tisztázzuk a hallgatóval, miért fontos ez a fejezet! Próbáljuk meg „elhitetni” vele, hogy szüksége van rá, hogy hasznos dolgokat fog tanulni. Soroljuk fel azokat a tudáselemeket, amelyeket megtanul majd a továbbiakban, ezzel előkészítjük arra, hogy megfigyelje a felsorolt szempontokat. Már elkezdjük a funkciók összekapcsolását, hiszen ha elolvassa a részecélkitűzést, bizonyos kifejezésekkel már ismerősként fog találkozni a tananyagban. Mindig egy jó tanáccsal és egy véletlenszerűen adott vicces idézettel, mondással zárom a célkitűzést, ami a kezdő feszültséget próbálja meg feloldani, hogy a „Tovább” gombra kattintva szívesen, vidám hangulatban kezdje a munkát.

Figyeljük meg, hogy az egyes funkcióval rendelkező részhez egy meghatározott ikon kapcsolódik. Érdeemes megszokássá tenni a hallgató számára, hogy ha az adott ikont látja, ahhoz valami azonnal kapcsolódjon az agyában. Készítsünk ikont a tananyagleíráshoz, a feladatokhoz, a tippekhez, az érdekességekhez, attól függően, hogy milyen jellegű tartalmat publikálunk a tananyagban.

A TANANYAG

A legnehezebb feladat maga a tananyag kifejlesztése. Hogy lehet olyan tananyagot elkészíteni, ami az adott célcsoportnak optimális? Nem túl egyszerű, de nem is megerőltető. Ha túl egyszerű ezért, ha túl nehéz, azért fogja elveszíteni a hallgató az érdeklődését. Vita zajlik például szakmai berkekben arról, hogy mekkora legyen egy weboldal mérete, amin a tananyag olvasható? Sokan azt mondják, hogy csupán annyi lehet, hogy a felhasználónak ne kelljen görgetnie. Mások szerint nyugodtan lehet 2-3 görgetésnyi oldal is. Azért tisztázzunk néhány fontos kérdést: mekkora egy oldal? Mennyi információt tartalmaz az az oldal? Mekkora a képernyő felbontása? Mekkora a hasznos terület? Hány kép található az oldalon? Milyen szintű a tananyag? Aki diktatórikusan kinyilvánítja a maximum egy oldal mennyiséget, nyilvánvalóan nem töpreng ilyen kérdéseken. Mert ugye nem mindegy, hogy alsó tagozatosoknak készül egy oldal, vagy érettségizetteknek. Nem ugyanazon mennyiség fér el 1024x768-es felbontású monitoron, mint 1680x1050-es felbontásban, ahogy az sem mindegy, hogy tömény jogi szövegről, vagy képekkel, ábrákkal tarkított laza szövegről beszélünk. S még nem tettem fel a kérdést: a felhasználó nem tud görgetni? Persze ez már csak rosszmaájúság volt, de bizony vannak olyan témakörök, amelyet nem lehet szétszedni több, apróbb al-lapra, egy lapon kell megtanítani mindent, akármennyi görgetéssel jár a dolog. Ez didaktikai kérdés, nem technikai. Természetesen jó dolog az, ha könnyen feldolgozható egységeket készítünk, ragaszkodjunk ehhez a módszerhez, de csak okosan, a lehetőségekhez mérten.

A tananyagfejlesztés esetén sokszor már meglevő, könyvben, jegyzetben levő tananyagot ültetünk át digitális formára. Ez nem rossz, viszont elvihet minket olyan irányba, ami esetleg gondot okozhat a későbbiekben. Az offline tananyagok általában tanárral történő feldolgozásra készülnek. Ha ezt nem vesszük figyelembe, és egyszerűen csak „digitalizáljuk” a tudáselemeket, akkor a hallgató maximum elolvassa a tananyagot, de instrukciók és az interaktivitás hiánya miatt nem fogja tudni optimálisan feldolgozni azt. Tehát mindenképpen szükséges úgy átalakítanunk a tudást, hogy a tanár feladatait is bele tudjuk foglalni. Ha mást nem, legalább írjunk egy kis instrukciót arra, hogy milyen sorrendben dolgozza fel a hallgató

a rendelkezésére álló anyagot, milyen feladatokat oldjon meg, s azokat hogyan tudja ellenőrizni. Bár már volt róla szó, ilyenek az *„Olvasd el és értelmezd a tankönyv 3. fejezetét, emeld ki a kulcsszavakat!”*, vagy *„A feldolgozás során gondolkodj el azon, hogy a saját háztartásodban milyen módon tudnád az energiafogyasztást csökkenteni. Készíts táblázatot a legtöbbet fogyasztó eszközökről, s készíts heti naplót, mennyi ideig használod őket!”* mondatok, instrukciók. Ezek valamilyen szinten pótolják a tanárt, segítenek a hallgatónak abban, hogy a megszabott irányvonalat követve tudják feldolgozni a digitális tananyagot. Természetesen a mikor és a hol kérdése a hallgatón múlik, éppen ezért kell olyan instrukciókat adni, hogy ne kelljen amiatt leállnia, mert az instrukciók nem megfelelőek, s az éjszaka közepén nem tudja elérni a tutort.

Használjuk ki a HTML adta lehetőségeket, hiszen bizonyos kiegészítésekkel interaktívvá tudjuk tenni az oldalt, vele együtt a tananyagunkat. A Javascript – bár csak kliensoldali nyelv – alkalmas arra, hogy apró interaktivitást csempésszünk az oldalakban. Ugyancsak használhatjuk a DHTML lehetőségeit, nem véletlenül hívják ezt a HTML kiterjesztést dinamikus HTML-nek. Csupán ötleteket tudok adni, hogy milyen irányba induljunk el, maga a megoldás mindig nagyban függ a feladattól, függ az oktató alkalmazott módszereitől. A következőkben láthatunk pár ötletet, forráskóddal, magyarázatokkal, hogy azok is megértsek ők, akik kevésbé járatosak a programozásban.

ELJÁRÁSOK ÉS FÜGGVÉNYEK

Amennyiben járatosak vagyunk a weblapszerkesztésben (miért ne lennénk, hiszen ezt akarjuk megtanítani a következőkben), valószínűleg használunk valamilyen oldalszerkesztő programot, mint pl. a Dreamweaver vagy a Frontpage. Ezek amennyire megkönnyítik a munkánkat, éppen annyira lehetetlenné teszik azt, ha mélyebben beleássuk magunkat a weblapokba. Nem igazán láttam még olyan profi weblapszerkesztőt, aki Frontpage-dzsel szerkesztette meg a PHP-s elemeit a weblapbank. Éppen ezért a dolgozatomban mindig a kézi bevitelű forráskódot részesítem előnyben, erre próbálom buzdítani a saját hallgatóimat is, ahogy az olvasót is szeretném megkérni arra, hogy ezt a módszert kövesse velem.

Kezdjük először a legegyszerűbbekkel: készítsük el az oldalak vázát. Már elkészítettük a forgatókönyvet. Láthattuk, hogy legalább kétféle lapra lesz szükségünk: egy fejezetlapra és egy tartalomlapra. Az első tartalmazza majd a fejezet instrukcióit, a második pedig magát a feldolgozandó tananyagot. Most még nem beszélünk a feladatokat, a feladatbankot vagy az önellenőrzést tartalmazó lapról, azt későbbre halasztjuk.

FEJEZETLAPOK

Elsőként gondolkodjunk el azon, hogy hogyan nézzen ki a fejezetlapunk. Szükségünk van egy célkitűzésre, a tudáselemek felsorolására, majd – bár ez csak nálam megszokott – egy vidám idézetcsíkra. Ehhez legegyszerűbb, ha DIV-eket definiálunk egy CSS-ben. Ehhez készítsünk egy content.css és egy f2.html állományt. Az első fogja tartalmazni a formai beállításokat, a második fájl pedig egy fejezetlap lesz, amiben teszteljük a stíluslapunkat.

A content.css tartalma egyelőre legyen az alábbi:

```
body {  
    background: #F2FFBF;  
    font-family: arial, verdana, helvetica, sans-serif;  
    font-size:13px;  
    text-align:justify;  
    color: black;  
    line-height:1.25em;  
    margin:0px;  
    padding-left:16px;  
    padding-right:16px;  
    padding-top:8px;  
}
```

Kezdeként beállítottuk a HTML oldal paramétereit, a háttér #f2ffbf színű lesz, Arial betűtípusú karakterekkel fogunk rá írni, 13px betűmérettel, fekete színnel, sorkizártan, 1.25-

os sortávval. A lapon nulla margót engedélyezünk, viszont balról és jobbról 16 pixelnyi, felülről pedig 8 pixelnyi távolságot tartunk. Próbáljuk is ki a megadott HTML lapon. Az f2.html-be illesszük be a következő kódot:


```
<head>
  <style type="text/css">
    @import url(content.css);
  </style>
  <title>2. fejezet</title>
  <meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-2" />
</head>

<body>

</body>
```

Ha elmentettük egy mappába mindkét állományt, akkor nyissuk meg egy böngészőben az f2.html-t. (Elég, ha Intézőben duplán kattintunk a fájlra.)

Ha jól csináltunk mindent, máris a megadott zöld színben pompázik az oldalunk. Természetesen, ha valakinek nem jön be ez a színösszeállítás, úgy változtatja a színeket, ahogy szeretné. Ha már eddig eljutottunk, írjunk is valamit a <body> tag-ek közé, mentünk, majd frissítsük az oldalt. Máris láthatjuk a lap szélétől való távolságot baloldalt és fent.



Hello World!

Eddig készen vagyunk az alappal, tegyük ki a fejezet címét az oldalra! Nálam ez *nodeTitle* néven fut, hozzuk létre őt is a CSS-ben:

```
.nodeTitle {
  color: #A1B386;
  font-size: 20px;
  font-weight: bolder;
  letter-spacing: 2px;
  padding: 2px;
  border-bottom: 2px solid #ACBF60;
  text-align: right;
}
```

Módosítottuk a <body> tag-hez képest a színt, a betűméretet, a stílust (félkövér lett), a betűközt, a távolságot a többi objektumtól, illetve húztunk egy vonalat is a szöveg alá, ami jobbra igazított lett. Mentünk, majd vegyük elő a HTML állományunkat. Nem elég, ha a

CSS-ben létrehozzuk a stílust, de használnunk is kell valahol. Legyen a <body> tag után az első sorunk a következő:

```
<p class="nodeTitle">A HTML alapjai</p>
```

A <p> tag segítségével új sorba kerül a feliratunk, a stílusa pedig nodeTitle lesz. Mentsünk, frissítsük az oldalt.

A HTML alapjai

Miért fontos, hogy kiírjuk a fejezetcímet? A hallgató előtt lesz, hogy éppen mit tanul. Jobbra igazítottuk, majd látjuk, hogy az egyetlen jobbra zárt elem a weblapon. Ez megint a megszokásokat erősíti, s mint tudjuk, a megszokások valamilyen szintű biztonságot nyújtanak. Nem hivalkodó, nem kiabál a felirat a weblapon. Próbáljuk ki nyugodtan pl. #ff00ff színnel, látni fogjuk, hogy egy erős kiemelés esetleg elvonhatja a hallgató figyelmét a fontosabb dolgokról. Ott van, nem tolakodó, de mindig a lap jobb felső sarkában nyújtja az információt.

Említettem már előzőleg, hogy a könnyebb tájékozódást a weblapon ikonokkal segítem. Ezek apró, nálam 60x60 pixel felbontású ikonok, ami lehetőleg stílusában illeszkedjen a tananyaghoz. Párat nézzünk meg, ötletként:



Ezek az ikonok egy idő után bevésődnek, könnyebb eligazodást biztosítanak. Hogy melyik ikon mire való, majdnem mindegy. Természetesen valamilyen szinten vonatkozzon a tevékenységre, pl. a hatodik ikon kiváló multimédiás célokra, de valószínűleg nem ezt fogjuk alkalmazni törzsanyag feldolgozására, vagy célmeghatározásra.

Hozzunk létre a következőkben egy olyan szövegdobozt, amelyben leírhatjuk a célkitűzésünket az adott fejezetre vonatkozóan. Most kezdjük a HTML résszel:

```

<div class="i_rck">
  
  <span class="iDeviceTitle">A tanulási egység
célkitűzései</span>

  <p>Az alábbi rész inkább elméleti jellegű lesz, de biztosan
tisztában vagy vele, hogy megfelelő mélységű elméleti ismeretanyag
nélkül sokkal nehezebb lesz a gyakorlatban megvalósítanod a
szükséges tennivalókat. Éppen ezért kérlek, hogy figyelmesen
olvasd el a tananyagot, figyelj a közben leírt instrukciókra, majd
legjobb tudásod szerint próbáld megoldani a tájékoztató lapok
végén található gondolkodtató feladatokat. Amennyiben kérdésed
merülne föl a megoldások közben, próbálj visszatérni a megfelelő
lapra, ha pedig ott sem találtad meg amit kerestél, vagy nem
egyértelmű a kérdés, fordulj bizalommal konzulensedhez.</p>

</div>

```

Az első `<div>` létrehozza az `i_rck` rész-célkitűzés dobozt, amely tartalmaz egy képet, egy címet és egy szöveget. Ehhez a CSS-ben szükségünk lesz 3 bejegyzésre: `i_rck`, `iDevice_icon` és `iDeviceTitle`. Hozzuk létre őket a CSS-ben:

```

.i_rck {
  background-color: #FAFFE6;
  padding: 8px;
  border: 1px solid #ACBF60;
  margin: 10px;
}

.iDevice_icon {
  padding-top: 0px;
}

.iDeviceTitle {
  color: #A1B386;
  font-size: medium;
  font-family: Arial, Helvetica, sans-serif;
  font-weight: bold;
  position: relative;
  top: -18px; /* igazítás az ikon közepére */
}

```

Picit világosabbra vettük a szövegdoboz hátterét, adunk neki egy keretet. Az ikon a bal felső sarokba kerül, a cím pedig az ikon közepére igazítva függőlegesen. Frissítés után a következő eredményt kapjuk, amennyiben az `iconz` mappába bemásoltunk egy 60x60-as `icon_objectives.gif` nevű GIF formátumú képet. Amennyiben más a mérete, más a formája a képnek, annak alapján módosítani kell a HTML kódot és a CSS-t is. Megvan? Akkor lássuk, mit kapunk eredményül:



A tanulási egység célkitűzései

Az alábbi rész inkább elméleti jellegű lesz, de biztosan tisztában vagy vele, hogy megfelelő mélységű elméleti ismeretanyag nélkül sokkal nehezebb lesz a gyakorlatban megvalósítanod a szükséges tennivalókat. Éppen ezért kérek, hogy figyelmesen olvasd el a tananyagot, figyelj a közben leírt instrukciókra, majd legjobb tudásod szerint próbáld megoldani a tájékoztató lapok végén található gondolkodtató feladatokat. Amennyiben kérdésed merülne föl a megoldások közben, próbáld visszatérni a megfelelő lapra, ha pedig ott sem találod meg amit keresel, vagy nem egyértelmű a kérdés, fordul bizalommal konzulensedhez.

Szerintem egyre szebbek leszünk! Figyeljük meg, hogy a speciális funkcióval rendelkező részeket egy kicsit kiemeljük, itt egy eltérő háttérszínnel. Érdekes használni a figyelemfelhívás ilyen módját, de vigyázzunk, ne vigyük túlzásba! Ha sokat emelünk ki, a hallgató elvész az oldalon, nem tudja majd eldönteni, mi mihez tartozik. Mutatok egy példát a káoszra:

The screenshot shows a website with a highly cluttered and inconsistent design. At the top, there's a search bar and a date announcement: "Ma 2010. 4. 4, vasárnap, Izidor napja van." Below this, a large banner features a building image and the text "Több mint 7 500 000 alkatrész... További részletek". To the right, a sidebar lists prices for digital products. The main content area is divided into several sections with different backgrounds and colors, including a blue bar with contact information, an orange bar with "OnLine Shop" and "Lejárt!" labels, and a central section for "Szórakoztató elektronikai és háztartási gépek". The bottom of the page has a navigation menu on the left with links like "Viszonteladónak", "Szolgáltatásaink", and "Webáruházak". On the right, there are two boxes for "Telefonos ügyelet" and "HelpDesk vonal". The overall layout is disorganized, with overlapping elements and a lack of clear visual hierarchy, illustrating the concept of "káosz" (chaos) in web design.

Sajnos, a mozgó részek nem látszanak, de bizony itt 4 rész is animálva van, ami egyrészt fölösleges, másrészt a káosz melegágya. Természetesen a cég nevét kivettem a képből, s bár

vásároltam már tőlük, nem is olyan egyszerű eligazodni még az olyan tapasztalt webes vásárlóknak sem, mint jómagam.

De mi a tökéletességre törekszünk, átgondoljuk a feladatainkat, figyelembe vesszük a hallgató igényeit is! Akkor folytassuk is, tegyük bele a tudáselemek ismertetését is. Ehhez én a Flash lehetőségeit vettem igénybe. Hogy miért nem egyszerűen kiírom a képernyőre, mint az előző blokkot? Elsősorban a fentiek miatt, azaz nem akarom többször ismételni magam. Másodsorban pedig nem árt, ha a hallgató figyelmét megragadja valami, mint itt a számítógép animációja. Vigyázzunk, ugyanis ha túl jóra sikeredik a figyelem megragadása, elvonhatjuk a figyelmét az oldal többi elemétől. Ezt úgy tudjuk orvosolni, hogy a lehető legkevesebb információt teszünk az oldalra. Kezdjük azzal, hogy beszúrjuk a Flash animációt!

```
<div style="text-align: center">
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash
.cab#version=6,0,40,0" width="680" height="550">
  <param name="play" value="false" />
  <param name="scale" value="exactfit" />
  <param name="src" value="animacio/2.0 a_html_alapjai.swf" />
  <param name="width" value="680" />
  <param name="height" value="550" />
  <embed type="application/x-shockwave-flash" play="false"
scale="exactfit" src="animacio/2.0 a_html_alapjai.swf" width="680"
height="550"></embed>
</object>
</div>
```

Egy középre igazított <DIV>-et használunk az animáció megjelenítésére, amelyben a szabványos Flash tag-eket használva megjelenítjük az adott animációt az *animacio* mappából, adott méretekkel. Ha megtekintjük az oldalt, láthatjuk, hogy a szokásos szabványszöveg után („E részcéltűzés megtanulása után...”) felsorolom a hallgató részére azokat a kompetenciákat, amit el fog sajátítani a feldolgozás során. Ügyeljünk rá, hogy egy részcéltűzés alkalmával ne soroljunk fel 5 célnál többet. Próbáljuk meg úgy felépíteni a forgatókönyvet, hogy inkább több, kisebb részből álljon. Gondoljunk arra, hogy a hallgató talán csak 10 percet tölt egy alkalommal a tananyaggal, nincs többre ideje. Sokan vannak például a játékokkal is így: azért nem kezdenek bele egy nagyobb, bonyolultabb játékba, mert nincs egyszerre annyi ideje. Inkább választja a pörgős, pár perces, jóval egyszerűbb játékokat.

A következő blokkal lezárjuk a weblapot. Teszünk egy hasonló szövegdobozt, mint a tetején, csak más tartalommal. A doboz definiálása ugyanúgy történik, a CSS-ben már létrehoztuk a szükséges formázásokat:

```

<div class="i_rck">

<table border=0>
  <tr>
    <td></td>
    <td nowrap valign=middle><b>A pillanat szüleménye: </b></td>
    <td class="itodMsg">
      <script>var l =
Math.floor(Math.random()*todmsg.length);document.write(todmsg[l]);
      </script>
    </td>
  </tr>
</table>

</div>

```

A fenti kód lényege a Javascript betét. Semmi mást nem tesz, csak véletlenszerűen generál egy számot 0 és egy *todmsg* nevű tömb hossza között, majd az ennek megfelelő tömbelemet kiírja. De honnan is jön ez a *todmsg* tömb? Egyszerűen létrehozuk! Hogy könnyebben tudjuk majd kezelni, egy különálló állományban hozzuk létre, amit majd beillesztünk a szükséges helyekre, például itt a HTML állományunkba. Szóval hozzunk létre egy *todmsg.js* nevű állományt, s kezdjük el feltölteni az alábbi módon:

```

todmsg=new Array();

todmsg[0] = "A cd-t tönkretettem, kérem a következőt."
todmsg[1] = "NYOMTATÓK: Miközben megpróbálta a módosításokat menteni, a következő hiba történt: A művelet sikeresen befejeződött. [OK]"
todmsg[2] = "MICROSOFT EXCEL: Nincs elég sor. [OK]"
todmsg[3] = "A FÁJL CSERÉJÉNEK JÓVÁHAGYÁSA: A fájl létezett, felülírtam! Nem baj? [IGEN] vagy [NEM]"
todmsg[4] = "TARTALMI TANÁCSADÓ: A tartalmi tanácsadó konfigurációs információja hiányzik. Valószínűleg valaki bütykölt vele. Ellenőrizze a tartalmi tanácsadó beállításait."
todmsg[5] = "Kérem, válasszon: [IGEN] [NEM] [NEMIGEN] [NEM TUDOM]"
todmsg[6] = "MICROSOFT CAR99 LÉGZSÁK HIBA: Vigyázat! Ön súlyos közlekedési balesetet szenvedett. Kívánja, hogy a légszák felfúvódjon?"
todmsg[7] = "BILLENTYŰZET NINCS INSTALLÁLVA: Windows 95 nem ismerte fel a billentyűzetet. Nyomd meg az [F1] gombot a továbblépéshez vagy az [F2]-t a megszakításhoz."
todmsg[8] = "A beírt jelszó nem hibás. Próbálja újra!"
todmsg[9] = "Töröltem a Windows-t! Baj? [igen] [nem] [mégsem]"
todmsg[10] = "HIBA: Nyomj [IGEN]-t vagy [MÉGSE]-t a továbblépéshez."

```

Látható, hogy a legelső sor definiálja a tömböt, a továbbiak pedig elkezdik feltölteni azt. Hogy kinek mekkora a türelme, s hogy milyen jellegű szövegeket ír bele, az már lényegtelen. Vagyis nem lényegtelen, csak legyen funkciója. Lehet a nap tippje, lehet egy vicces mondat (mint itt), de írhatunk bele bölcsességeket is, vagy egy-egy kulcsszó definícióját, ami az előző fejezetben előfordult. A lehetőségek csak a fantáziánkon múlnak.

Most hogy lementettük a fájlt, illesszük be HTML állományba! A <head> részt bővítsük ki egy sorral:


```
<script type="text/javascript" src="todmsg.js"></script>
```

Ezzel beolvassuk az ugyanabban a mappában található állományt. Már csak egy dolog hiányzik, a hivatkozott „itodMsg” stílus létrehozása a CSS-ben. Legyen, mondjuk az alábbi:

```
.itodMsg {  
    color: #408040;  
    font: bold 16px Arial;  
    line-height: 1.25em;  
    text-align: left;  
    padding-left: 8px;  
}
```


Ha ikonunk is van már a megadott helyen, próbáljuk ki a lapot! Meg fog jelenni a lap alsó részén a szövegdobozunk, s benne egy sor a *todmsg.js*-ben leírtak közül. Lássuk egyben az egészet!

A HTML alapjai



A tanulási egység célkitűzései

Az alábbi rész inkább elméleti jellegű lesz, de biztosan tisztában vagy vele, hogy megfelelő mélységű elméleti ismeretanyag nélkül sokkal nehezebb lesz a gyakorlatban megvalósítanod a szükséges tennivalókat. Éppen ezért kérek, hogy figyelmesen olvasd el a tananyagot, figyelj a közben leírt instrukciókra, majd legjobb tudásod szerint próbáld megoldani a tájékoztató lapok végén található gondolkodtató feladatokat. Amennyiben kérdésed merülne föl a megoldások közben, próbáld visszatérni a megfelelő lapra, ha pedig ott sem találsz meg amit kerestél, vagy nem egyértelmű a kérdés, fordulj bizalommal konzulensedhez.




Webmester E-learning

E rész célkitűzés megtanulása után az alábbi tudáselemekkel leszel gazdagabb:

- 1) Megtudod, milyen eszközökre, szoftverekre lesz szükséged a fejlesztéshez.
- 2) Megismered, hogyan kommunikál a böngésződ a webszerverrel.
- 3) Látni fogod, mit jelent az, hogy a HTML egy "lapleíró" nyelv.
- 4) Megtanulsz HTML kódot beírni.
- 5) Le fogod tudni tesztelni a megírt HTML kódot.

Jó munkát kívánok!



A pillanat szüleménye: **MICROSOFT EXCEL: Nincs elég sor. [OK]**

TARTALOM OLDAL

Miután kiváló fejezetlapot készítettünk, picit nehezebb feladat következik. El kell készítsük a tanulási egységeket tartalmazó oldalak vázát. Először gondolkodjunk el azon, mit fognak ezek az oldalak tartalmazni! Biztosan lesz törzsanyagunk. Ez fogja leírni a tudáselemeket. Aztán, mivel HTML-t oktatunk, szükségünk lesz attribútumleírásra. Itt fogjuk leírni a tanult tag-ek szintaxisát. Lesznek forráskódjaink is, hiszen példákat kell mutassunk a hallgatóknak. Előfordulhatnak olyan szavak, kifejezések, amelyek magyarázatra szorulnak. Milyen jó lenne, ha nem külön szöszedetben, hanem közvetlenül a tanult helyen már magyarázatot kapna a hallgató a kifejezésre! Aztán lesznek biztosan feladatok. Ezen belül gondoskodnunk kell olyanokról, amelyek csupán leírják az elkészítendő munkát, de lennie kell olyannak is, amely letölthető segédleteket biztosít (pl. képek, nyers szövegek, videók), sőt, nem ártana, ha egy lehetséges megoldást is megtekinthetnénk, motiválandó a hallgatót új megoldások, utak keresésére. Ez így nagyon sok volt egyszerre, de látni fogjuk, hogy elmondani hosszabb, mint megoldani. Mivel is kezdjük? Hozzunk létre egy sablont, majd készítsünk egy olyan oldalt, amely törzsszöveget tartalmaz, némi szómagyarázattal!

ALAPOK

Legyen a fájlunk neve *f2_1.html*. Ezzel utalunk arra, hogy a második fejezet, első részecéljának kidolgozását tartalmazza a lap. A `<body>` tag-ig már ismerjük a szükséges sorokat. Behívjuk a *content.css*-t, s beállítjuk a kódlapot. Most nem lesz szükségünk a Javascript részre, de hagyjuk benne, később felhasználjuk. A `<body>` tag után folytathatjuk az oldal létrehozását. Elsőként írjuk ki a fejezet címét a már ismert *nodeTitle* stílussal. Ezek után viszont szükségünk lesz arra is, hogy az aktuális részecélkitűzés címe is megjelenjen. Tájékoztassuk a hallgatót arról, hogy éppen hol jár. A módszer ugyanaz, mit az előző sorban, egy `<p>` tag stílusát változtatjuk meg, hogy megfelelő legyen számunkra:

```
<body>

<p class="nodeTitle">Előkészületek</p></div>
<p class="bottomTitle">Mire lesz szükségünk a fejlesztéshez?</p>
```

A fejezet címe tehát „Előkészületek”, a részecélkitűzése pedig „Mire lesz szükségünk a fejlesztéshez?”. Természetesen nem feledjük létrehozni a *bottomTitle* stílust sem!

```
.bottomTitle {
  color: #A1B386;
  font-size: 20px;
  font-weight: bold;
  letter-spacing: 2px;
```

```
padding: 0px;  
text-align: left;  
}
```

Semmi fontos dolgot nem tettünk, csak balra igazítottuk a feliratot. Készen vannak a címfeliratok, jöhet maga a tananyag. A legegyszerűbb utat választom, azaz beillesztem az aktuális célkitűzés tényleges szövegét, elejétől a végéig. Utána pedig feldogozzuk, megnézzük, mit kell még elkészítenünk, hogy a tartalom az általunk kívánt formában és interakcióval jelenjen meg. Szóval a tartalom:

```
<div style="display:block;">  
<p class="i txt">  
Az interneten bolyongva gyakran találkozunk jobbnál jobban elkészített  
weblapokkal. Gondolom, ezt a szöveget is azért olvasod, mert szeretnél Te  
is hasonlókat készíteni. Nos, ebben a fejezetben megismerkedhetsz a HTML  
alapjaival. Ha megtanulod az itt leírt dolgokat (no meg akövetkező  
fejezetekben sem csüggedsz!), Te is megjelenhetsz kreálmányoddal a  
világhálón. Persze léteznek ilyen-olyan HTML editorok (szerkesztők,  
készítők), melyekkel könnyebben el tudnád végezni a feladatot. De azért  
jó, ha megtanulod azt, hogy miből is épülnek fel a weblapok, mert így  
mindenféle komolyabb programok használata nélkül is készíthetsz  
weblapot.<br />  
A HTML egy angol betűszó, teljes alakban <b>HyperText Markup  
Language</b>, vagyis hipertext leíró nyelv. Ez az a nyelv, amellyel  
készítették azt a rengeteg weblapot, mely az interneten fellelhető. Az  
már más kérdés, hogy komolyabb programokkal is készítenek weblapot. De  
azok is eme nyelv parancsait használják. Úgyhogy szerintem jól jársz, ha  
megtanulod a HTML parancsait.  
<br /><br />  
Amikor megnézünk egy jó filmet, magával ragad minket a történet, a  
látvány, mindaz a forgatag, amit a film készítői készítettek nekünk. Ha  
jól végezték a munkájukat, észre sem vesszük azt a hatalmas munkát, amely  
a színpad mögött, akár több évig is zajlott: színészek munkája,  
kaskadőrök, digitális technika, blue- (green-) box technika,  
megvilágítások, stb. Fel sem tudjuk sorolni azokat a trükköket, amiket a  
végeredmény minőségének érdekében felhasználnak.  
<br />  
Ugyanez zajlik egy weboldal készítésekor is. Ami a látogató számára  
megjelenik, az csak a végső produktum. Hogy milyen hatalmas munka fekszik  
az egészben, talán nem is látszik, ezt csak az tudja, aki mindezért  
megdolgozott.  
Egy webszerkesztő feladata komplex: ő a weblap rendezője,  
forgatókönyvírója, gyártásvezetője, az összes színésze, menedzsere. E  
munkák ellátásához megfelelő eszközökre van szükség. Ilyen eszközök:  
HTML, CSS, JavaScript, PHP, SQL. A történet ezekről szól.  
</p>  
  
<p>No, de most nézzük, mik is kellene a gyakorlatban egy weblap  
elkészítéséhez.</p>  
  
<ul>  
  <li>Egy szövegszerkesztő. Igen. Akármilyen hihetetlen is, de itt a  
legalapabb szövegszerkesztők használata javasolt. Ilyen pl. a Windows  
alatt a Notepad, vagyis a Jegyzettömb, valamivel profibb már a Notepad++  
nevű, amely szintén ingyenesen letölthető. (Legutolsó lelőhelye a<span  
style="color:#ff6600"> http://notepad-  
plus.sourceforge.net/hu/site.htm</span> weboldal volt, de ajánlom  
megkeresésére a google-t!) Más operációs rendszereken az ehhez hasonló  
kaliberű szerkesztőket ajánlom.</li>
```

```

<li>Egy internetböngésző, amellyel elkészült művedet megnyithatod,
tehát tesztelheted, ellenőrizheted.</li>

<li>És persze némi tudás. Ez a fájl fedezi az alapismeret-
bevitelt. Vagyis kezdetnek ez a részletes, érthető szöveg maximálisan
megfelel. </li>
</ul>

<p>És ami NEM szükséges:</p>

<ul>

    <li>Internetkapcsolat. <a href="#" onmouseout="hidetip()"
onmouseover="tip('Kapcsolat nélküli mód',300)" class="tips">Offline</a>
módban is tudod tesztelni a weblapodat. (Kivéve a teljes
URL-es linkeket. Erről később.)</li>

    <li>Komoly szövegszerkesztő vagy <a href="#"
onmouseout="hidetip()" onmouseover="tip('Olyan program, amely
segítségével WYSIWYG módban tudsz weboldalakot tervezni.', 300)"
class="tips">HTML editor</a>. Előbbi kifejezetten nem ajánlott, utóbbira
pedig semmi szükség. </li>

</ul>

<p class="i_txt">Kis kedvcsinálóként nézzünk meg <a href="2.1.5
pix.html">pár képet</a> a profi munkák közül!</p>

<p class="i_txt">
Ha pedig megtanultuk az alapokat, érdemes továbblépni. Némi <a href="#"
onmouseout="hidetip()" onmouseover="tip('Az Adobe (volt Macromedia) igen
népszerű vektoralapú fejlesztőrendszere. Elsősorban animációk és
felületek készítésére használják a weblapokon.',300)"
class="tips">Flash</a> tudással, no meg rengeteg grafikai érzékkel
gyönyörű weboldalakot készíthetünk:
</p>

<table align="center" border="0">

    <tbody>

        <tr>

            <td><a href="2.1.1 flash1.html"></a></td>

            <td><a href="2.1.2 flash2.html"></a></td>

        </tr>

        <tr>

            <td><a href="2.1.3 flash3.html"></a></td>

            <td><a href="2.1.4 flash4.html"></a></td>

        </tr>

    </tbody>

</table>

```

```
</body>
```

Az első rész főleg szövegblokkokból áll `<p>` tag-ek közé zárva. Amit viszont észrevehetünk az az, hogy a notepad++ linkje hibásan van beillesztve:

```
<span style="color:#ff6600"> http://notepad-  
plus.sourceforge.net/hu/site.htm</span>
```

Nos, ez valóban nem a megszokott `<a>` tag, de máris kimagyarázom magam! Ne tegyünk a törzsanyagot tartalmazó oldalakra élő linket! Ne engedjük, hogy a hallgató kilépjen a keretrendszerből! Ha nagyon muszáj linket használnunk, illesszük be úgy, hogy ne lehessen rá kattintani, de lehessen másolni. Amit viszont érdemes: a linkeket egy lapon, a képzés végén összegyűjteni, de akkor sem mindegy, hogy hogyan, ajánlott egy meghatározott nevű lapra irányítani a *target* paraméter segítségével. Szóval nincsenek élő linkek, csak ha a saját rendszeren belül kapcsolódik valahová. Ez máris látható, ahogyan az „Offline” szavunkra nézünk. Fekete a szín, de narancs, szaggatott vonallal van aláhúzva. Ez bizony egy link. De nem is akármilyen! Ez a szómagyarázó link. Ha ebben az állapotban leteszteljük, vagy megnézzük a forrást, láthatjuk, hogy igazából sehová sem akar menni, a cél helyére egy „#” került, ami azt fogja jelenteni a böngészőnek, hogy nem kell mennie sehová.

```
<a href="#" onmouseout="hidetip()" onmouseover="tip('Kapcsolat nélküli  
mód',300)" class="tips">Offline</a>
```

Van viszont neki két eseménye és egy stílusa. Az első esemény a *hidetip()*, ami akkor lép működésbe, ha *onMouseOut* esemény következik be, azaz ha a linkről elvisszük a kurzort. A nevéből ítélve ekkor elrejtí a tippet, azaz a szó magyarázatát. Az *onMouseOver* esemény pontosan az ellenkezője: akkor aktiválódik, ha ráállunk az egérrel a linkre. Ekkor jelenjen meg egy ablak, ami tartalmazza a „Kapcsolat nélküli mód” magyarázószöveget, a 300 pedig az ablak szélessége lesz. Foglalkozunk össze: van egy linkünk (az Offline szó), amire ha ráállunk, megjelenik egy magyarázó ablakocska, ha elmegyünk róla az egérrel, akkor eltűnik. Szép, szép, de meg is kell ám oldani. Akik jártasak a Javascript és a DHTML rejtelseiben, azoknak ez csupán ujjgyakorlat. Akik meg nem, azok használják az alábbi scriptet, amit sok-sok magyarázattal láttam el, hogy könnyebben lehessen akár testreszabni, vagy megérteni.

A scriptet mentjük le *tip.js* néven, majd az előzőleg megmaradt (ugye nem töröltük ki?) *todmsg.js*-t írjuk át *tip.js*-re!


```

var offsetfromcursorX=12 //x eltolás az aktuális kurzorpozíciótól
var offsetfromcursorY=10 //ugyanaz y-ra

var offsetdivfrompointerX=10 // x eltolás a DIV-hez képest
var offsetdivfrompointerY=14 //ugyanaz y-ra

document.write('<div id="dhtmltooltip"></div>') //kiírjuk magát a DIV-et
document.write('') //Ez a
div teteje, egy nyíl, az iconz mappából

// jöhetnek a böngészővizsgálatok, sajnos, szükséges
var ie=document.all
var ns6=document.getElementById && !document.all
var enabletip=false
if (ie||ns6)
var tipobj=document.all? document.all["dhtmltooltip"] :
document.getElementById? document.getElementById("dhtmltooltip") : ""

var pointerobj=document.all? document.all["dhtmlpointer"] :
document.getElementById? document.getElementById("dhtmlpointer") : ""

function ietruebody(){
return (document.compatMode && document.compatMode!="BackCompat")?
document.documentElement : document.body
}

// ez az a script, amelyik kiírja az ablakot, 3 paramétere van: szöveg,
szélesség, szín
function tip(thetext, thewidth, thecolor){
tipobj.style.width="300px";
tipobj.style.padding="4px";
if (ns6||ie){
if (typeof thewidth!="undefined") tipobj.style.width=thewidth+"px"
if (typeof thecolor!="undefined" && thecolor!="")
tipobj.style.backgroundColor=thecolor
tipobj.innerHTML=thetext
enabletip=true
return false
}
}

function positiontip(e){
if (enabletip){
var nondefaultpos=false
var curX=(ns6)?e.pageX : event.clientX+ietruebody().scrollLeft;
var curY=(ns6)?e.pageY : event.clientY+ietruebody().scrollTop;
//kiszámoljuk, az egér milyen közel van az ablak sarkához, hgy lehetőleg
ne csússzunk ki
var winwidth=ie&&!window.opera? ietruebody().clientWidth :
window.innerWidth-20
var winheight=ie&&!window.opera? ietruebody().clientHeight :
window.innerHeight-20

var rightedge=ie&&!window.opera? winwidth-event.clientX-offsetfromcursorX
: winwidth-e.clientX-offsetfromcursorX
var bottomedge=ie&&!window.opera? winheight-event.clientY-
offsetfromcursorY : winheight-e.clientY-offsetfromcursorY

var leftedge=(offsetfromcursorX<0)? offsetfromcursorX*(-1) : -1000

//ha nem elég a vízszintes hely
if (rightedge<tipobj.offsetWidth){
//akkor áttesszük a szemközti oldalra
tipobj.style.left=curX-tipobj.offsetWidth+"px"
nondefaultpos=true

```



```

}
else if (curX<leftedge)
tipobj.style.left="5px"
else{
//hol is van az egér pozíciója (x)
tipobj.style.left=curX+offsetfromcursorX-offsetdivfrompointerX+"px"
pointerobj.style.left=curX+offsetfromcursorX+"px"
}

//egér y
if (bottomedge<tipobj.offsetHeight){
tipobj.style.top=curY-tipobj.offsetHeight-offsetfromcursorY+"px"
nondefaultpos=true
}
else{
tipobj.style.top=curY+offsetfromcursorY+offsetdivfrompointerY+"px"
pointerobj.style.top=curY+offsetfromcursorY+"px"
}
tipobj.style.visibility="visible"
if (!nondefaultpos)
pointerobj.style.visibility="visible"
else
pointerobj.style.visibility="hidden"
}
}

// nem kell az ablak, kirakjuk -1000-re, és el is rejtjük a biztonság
kedvéért.
function hidetip(){
if (ns6||ie){
enabletip=false
tipobj.style.visibility="hidden"
pointerobj.style.visibility="hidden"
tipobj.style.left="-1000px"
tipobj.style.backgroundColor=''
tipobj.style.width=''
}
}

// inicializáljuk a scriptet
document.onmousemove=positiontip

```

Sajnos, nem túl elegáns a kód, de mindenre figyelni kell. A böngészők közötti különbségek (amelyek azért már kezdenek lassan eltűnni) meg fogják keseríteni az életünket. Folyamatosan figyelni kell, hogy milyen böngészője van a hallgatónak, s még akkor sem biztos, hogy két böngészőn ugyanúgy fog kinézni az oldalunk. De ne keseregjünk, inkább haladjunk tovább.

Ha valaki tüzetesen átböngészte a script kódját, észrevehette, hogy CSS hivatkozásokat tartalmaz: dhtmltooltip, dhtmlpointer. Ezek nélkül használhatatlan a scriptünk, mivel el kell készítenünk az ablakokat. Írjuk bele hát a CSS-be az alábbi definíciókat:

```

/* Szómagyarázat stílusai */
#dhtmltooltip{
position: absolute;
left: -300px;
width: 150px;
border: 1px solid black;
padding: 2px;
background-color: #ffff80;
visibility: hidden;
z-index: 100;
font: normal 8pt Arial;
}

#dhtmlpointer{
position: absolute;
left: -300px;
z-index: 101;
visibility: hidden;
}

a.tips { color: #000000;
text-decoration: none;
border-bottom: 1px dashed #FF6600;
}

a.tips: hover {
color: #FF6600;
}

```

Láthatjuk a kódban *tips* stílus definícióját is, amit a HTML részben használtunk. Itt adtuk meg, hogy szaggatott, fekete színű legyen a linkünk. A *hover* tulajdonság pedig átszínezi a linket, ha ráállunk az egérrel, megkapjuk a narancs árnyalatot is. Ha mindent jól csináltunk, valami ilyesmit kapunk:

És ami NEM szükséges:

- Internetkapcsolat. Offline módban is tudod tesztelni a weblapot. (Kivéve a teljes URL-es linkeket. Erről később.)
- Komoly szövegszerkesztő vagy HTML editor. Előbbi kifejezetten nem ajánlott, utóbbira pedig semmi szükség.

Kis kedvcsinálóként nézzünk


Kapcsolat nélküli mód

Bárhol használhatjuk a scriptet a szövegen belül, a megadott szintaxissal.

A lap végén négy kép található, amely négy másik lapra hivatkozik, ahol példákat láthat a hallgató a „jó” weboldal kategóriákra. Érdekes elámítani a hallgatót, ezzel is motiválva van, hogy esztétikailag és technikailag hasonló weblapokat hozzon létre. A teljes oldal megtalálható a [\[2\]](#) linken. Próbáljuk ki a Flash lehetőségeket, különösen ajánlom a „pár képet” linket, amely biztosan tetszik majd a diákoknak.

HALADÓ MEGJELENÍTÉS

Akármennyire is szép ez az oldal, még hiányoznak részek. Ilyen például az attribútumleírás, vagy a forráskód megjelenítés. Az elsőt nem bonyolítjuk túl, hasonlóan oldjuk meg, mint a célkitűzéseket. Létrehozunk egy ikonos címet, majd alá felsoroljuk az attribútumokat.



<BODY> attribútumleírás

text	A szöveg színét adhatjuk meg a szín angol nevével vagy hexadecimális kóddal.
link	A még nem látott linkek színét lehet megadni a szín angol nevével vagy hexadecimális kóddal.
vlink	A már látott linkek színét lehet megadni a szín angol nevével vagy hexadecimális kóddal.
alink	A megadott színűvé válik a link, ha éppen rákattintunk. Itt is a szín angol nevét vagy hexadecimális kódot alkalmazhatunk.
bgcolor	A háttérszínt adhatjuk meg a szín angol nevével vagy hexadecimális kóddal.
background	Itt értéként megadhatjuk egy képfájl elérési útját. Amikor valaki megnyitja az oldalt, a böngésző a megadott képpel telepakolja a weblapot, tehát addig rakja egymás mellé és alá, míg ki nem tölti teljesen az oldalt.

Hasznos lehet táblát alkalmazni az attribútum és a magyarázat elválasztására, s máris olvashatóbb, áttekinthetőbb lesz a leírás.

A forráskód megjelenítés már sokkal bonyolultabb művelet. Nem ragaszkodunk a helyes szintaxisleíráshoz, tehát nem profi szinten fogjuk megjeleníteni a forrást. Annyit teszünk, hogy az idézőjelbe kerülő dolgokat kékkel, a szövegeket zölddel, az összes többit pedig bordó színnel fogjuk megjeleníteni. A forráskód ablaknak fontos jelentősége lesz: le lehet majd őket futtatni! Ezért teszünk egy ikont a forráskód sarkába, amire kattintva, új ablakban megjelenik az eredmény. Nézzük meg inkább a képernyőképet, érthetőbbé teszi a magyarázatot.

```
<HTML>

<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=ISO-8859-2">
</HEAD>

<BODY bgcolor="#FFFC0" background="images/hatter.jpg" text="#FF0000" link="#0000FF" vlink="#00FF00"
alink="#0000FF">
  Ez a HTML oldalunk törzse
</BODY>

</HTML>
```



A fenti példa a <body> tag használatát mutatja be, illetve az oldal felépítését példázza: <html> és <head> tag-ek. Maga a Javascript forrás karakterenként, írógépszerűen megjeleníti a kódot. Sajnos, a megjelenítendő forrást nem egyszerű átadni egy Javascript rutinnak. Ha az eredeti HTML kódot adjuk át, és jelenítjük meg, akkor a kiírt forrás

végrehajtódik, de nem jelenik meg. Ezt még fokozza a sorok előtti tabulátorok kiírása, és a színezés. Ezért speciális trükköt alkalmazunk, a HTML részére nem értelmezhető módon adjuk át a forrást. Egy példával jobban érthetővé válik. Íme a kód:

```
<div align=center>
<div id="zoom"></div>
<div id="notepad" style="width:740px; height:200px"></div>
<div id="contentToWrite"
style="display:none">[HTML]××[HEAD]××[META http-equiv="Content-
Type" content="text/html; charset=ISO-8859-2"]×[/HEAD]××[BODY
bgcolor="#FFFFC0" background="images/hatter.jpg" text="#FF0000"
link="#0000FF" vlink="#00FF00" alink="#0000FF"]××Ez a HTML
oldalunk törzse×[/BODY]××[/HTML]</div>
</div>
<script type="text/javascript">writeContent(true);</script>
</div>
```

Láthatunk két speciális kódot: „×” és „×”. Feltehetőleg ezek a karakterek nem fordulnak elő a forrásban, de természetesen ez megváltoztatható. Az első jel az új sor jelzésére szolgál a kiíráskor, a második pedig egy tabulátort jelent. Mindkettőből akármennyit lehet halmozni, amire szükségünk is lesz. Láthatjuk, hogy a „<”, „>” jelek helyett szögletes zárójeleket írtam, ez megint arról szól, hogy a kiértékelő, megjelenítő, ne ismerje fel a HTML tag-eket.

Látható, hogy szükségünk lesz némi módosításra a CSS-ben is, no meg egy újabb Javascript beszúrására. Lássuk a szükséges módosításokat, elsőként a CSS-t, ez a könnyebbik rész!

```
/* forráskód megjelenítő */

#zoom {
    color:red;
}

#notepad, #notepad blink{
    width:500px;
    height:200px;
    background:white;
    color: #008000;
    font:12px Georgia;
    border:2px solid #ACBF60;
    text-align:left;
    padding:4px;
}

blink{
    display:inline;
}
```

A következő lépés maga a megjelenítő Javascript kódja. Nálam ez *typewriter.js* néven szerepel, a tartalma pedig a következő:

(45)

```

        if (theChar=="\"") { idjel++; if (idjel%2==1) theChar="\"<font
color=#0000ff>"; else theChar="</font>\""; }

        initString = initString + theChar + "<SPAN id='blink'>|</SPAN>";
        document.getElementById("notepad").innerHTML = initString;

        charIndex = charIndex/1 +1;
        if(charIndex%2==1){
            document.getElementById('blink').style.display='none';
        }else{
            document.getElementById('blink').style.display='inline';
        }

        if(charIndex<=stringLength){
            setTimeout('writeContent(false, '"+idjel+"' ',10);
        }else{

            obj = document.getElementById('notepad');
            w = parseInt(obj.style.width);
            h = parseInt(obj.style.height);
            l = obj.offsetLeft;
            t = obj.offsetTop;

            var ver = navigator.appVersion;
            var zx=(ver.indexOf("MSIE") != -1)?4:-8;
            var zy=(ver.indexOf("MSIE") != -1)?4:-8;

            zoom.style.position = "absolute";
            zoom.style.left = (parseInt(l+w-80-zx));
            zoom.style.top = (parseInt(t+h-63-zy));

            blinkSpan();

        }
    }

    var currentStyle = 'inline';
//villogás
    function blinkSpan(){
        if(currentStyle=='inline'){
            currentStyle='none';
        }else{
            currentStyle='inline';
        }
        document.getElementById('zoom').style.display = currentStyle;
        setTimeout('blinkSpan()',500);
    }
}

```

Nem azt mondom, hogy a legszebb, legtömörebb kód, sőt azt sem, hogy mindenképpen szükséges. Láttam olyan kész kódokat, amely egyszerűbben oldotta meg a fenti problémát, például egy megadott külső állományt olvasott be, s jelenítette meg hasonlóképpen. Hogy miért írtam inkább saját eljárást? Egyrészt nem akartam Flash-t használni, ha nem muszáj. Másrészt ez a script könnyen testreszabható, átírható, nem úgy, mint egy befordított SWF állomány. Van némi interaktivitás ebben is (például a fent említett Flash nem tudta

megmutatni a forrás eredményét!), s minden digitális tananyagban érdemes arra törekedni, hogy messzemenőig testreszabható legyen az egész rendszer.

FELADATOK

Már csak egyetlen funkciót szeretnék a tartalom sablonok témakörben megismertetni. Ez pedig a feladat/megoldás blokk. Mikor tananyagot készítünk, az elején el kell döntenünk, hogy milyen jellegű feladatokat fogunk adni a hallgatóinknak. Bontsuk három részre a feladatok típusait:

1. A törzsanyag közben adott feladatok alkotják az első csoportot. Ezek a feladatok elsősorban elgondolkodtató, műveltető jellegűek, a tananyag jobb megértését célozzák, s nem az ellenőrzés a céljuk. Nyugodtan megadhatjuk a megoldásokat is, de legtöbbször olyan feladatok tartoznak ide, amelyeknek nincs is megoldásuk, maximum némi segítséget, instrukciót tudunk a hallgatónak nyújtani. Ilyenek például a „Gyűjtsön adatokat...”, „Figyelje meg...” jellegű feladatok. Ezekben a feladattípusokban redundáns információkat, érdekességeket adhatunk át a hallgatónak (ezek az apró betűs részek a tankönyvekben), hatásuk motiváló lehet, de önellenőrzési célokat is elláthatnak.
2. A második típusa a feladatoknak a tananyag, a fejezetek, esetleg a részcélkitűzések végén található, ellenőrzési, önellenőrzési feladatok. Konkrét értékelés kell tartozzon hozzájuk, lehetőleg ölelje fel az adott rész teljes kompetenciáját. Ha motiválni szeretnénk a hallgatóinkat, tegyük ezt is interaktívvá: mindenképpen szólítsuk meg, adjuk meg az időkeretet, s a végén értékeljünk. Az értékelés ne csak „5 pontot értél el a 10-ből” legyen, de adjunk neki tanácsokat, hogy ez milyen szintnek felel meg az elvárthoz képest, s mit tehet annak érdekében, hogy jobb eredményt érjen el.
3. Az utolsó típus a feladatbank típusú feladatok. Ezek nem kötelező jellegű feladatokat tartalmaznak, fejezetekre bontva. A feladatok nehézsége legyen változó, legyen nagyon könnyű, de akadjon közöttük olyan is, amely esetleg nem megoldható az adott fejezet törzsanyagának birtokában. Legyen igénye a hallgatónak, hogy ha nem tudja eddigi tudásával megoldani a feladatot, tekintsen ki a tananyagból, próbáljon meg külső információs forrásból származó tudáselemeket is beépíteni tudásába. Ne feledkezzünk el a megoldásokról sem! Minden esetben adjunk meg egy lehetséges megoldást, amit érdemes kihangsúlyozni, hogy „egy lehetséges” megoldás. Adjunk lehetőséget a hallgatónak (bár ez már szervezési probléma), hogy amennyiben elakad a saját megoldásában, tudjon tanácsot kérni, ha pedig saját megoldással állt elő, el tudja küldeni azt az instruktorának, aki megfelelő módon értékelni is tudja azt.

TÖRZSANYAG FELADATAI

Készítsünk mindegyik típusra egy-egy példát! Az első változatra a táblázatok témakörből vettem a feladatot. A feladatunk a következő: adok egy forráskódot, a hallgató feladata, hogy kitalálja, milyen táblázatot fog eredményezni az adott forrás. Ehhez biztosítanunk kell magát a forrást, majd a megoldást. A forráskódok sok esetben nagy terjedelműek, nem lenne hasznos, ha a törzsanyag területéből vennének el részeket. Éppen ezért megadhatjuk úgy is, hogy csak akkor jelenjen meg a forrás, ha a hallgatónak szüksége van rá. Ehhez egy egyszerű DHTML scriptet fogunk alkalmazni, amely le- és felgördíti az adott DIV-et.

```
function insert_span(fa,eredmeny,felirat,szabad)
{
    if (!felirat) felirat="Megoldás";
    if (!szabad) szabad="0";
    eredmeny = eredmeny.replace(/x/g,"<br>"); // az x legyen ENTER
    eredmeny = eredmeny.replace(/ /g,"&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"); // a szó
        pedig legyen TAB (most elég 3 space)
    eredmeny = eredmeny.replace(/\[/g,"&lt;"); // nyit és majd a zár
    eredmeny = eredmeny.replace(/\]/g,"&gt;");

    document.write("<span id=vSec"+fa+" title="+szabad+"
style=\"cursor:hand; position:relative; display:none\"></span>");
    document.write("<span id=cSec"+fa+" style=\"cursor:hand; position:
relative; display:all\"><br>");
    document.write("<a onclick=\"return
_dSectionExpand('"+fa+"');\"><img src='iconz/expand.gif' border=0
width=16 height=16>&nbsp;&nbsp;&nbsp;");
    document.write("<b>"+felirat+"</b></a></span>");
    document.write("<span id=xSec"+fa+" style=\"cursor:hand;
position:relative; display:none\"><BR>");
    document.write("<a onClick=\"return
_dSectionCollapse('"+fa+"');\"><img src='iconz/collapse.gif' border=0
width=16 height=16>&nbsp;&nbsp;&nbsp;");
    document.write("<b>"+felirat+"</b></a><br><br><table width=80%
border=0 cellpadding=10><tr><td width=2></td><td
style=\"border-style: dashed; border-width:1; border-color:#ACBF60;
background-color:#f8f8ff;\"><span style=\"font:normal 8pt Verdana;
color:#304060;\">"+eredmeny+"</span></td></tr></table>");
    document.write("</span></span>");
}
```

Az első eljárásban négy paraméter felhasználásával tesszük ki a rejtett DIV-et. Az első paraméter egy azonosító, ami nálam az adott oldalon a feladat sorszámát jelenti. Vigyázzunk, ha ugyanazon sorszámot adjuk, kellemetlen következményei lehetnek! A második paraméter a legördítés eredménye, azaz maga a tartalom. Használhatjuk az előzőekben már tárgyalt karaktereket annak elkerülésére, hogy a böngésző értelmezni próbálja a HTML tag-eket. A felirat az a szöveg, amire a hallgatónak kattintani kell, hogy megjelenjen az eredmény, a szabad pedig egy bináris értékű paraméter, amely megmondja, hogy szabad-e látnia a hallgatónak az eredményt. Ezt majd később használni fogjuk! Addig is szúrjunk be egy feladatot az oldalba a fenti függvény használatával:


```
<script>insert_span(1,"<img  
src=\"images/table_feladat.png\">","Forráskód",1);</script>
```

A fenti sor nem tesz mást, mint egy képet szúr be a weboldalra, amely csak a „Forráskód” szövegre kattintva jelenik meg. (Mivel az a feladat, hogy az adott forráskódból kell kitalálni a hallgatónak, hogy mi lesz az eredménye, nem baj, ha nem szövegként adjuk meg a forrást, hiszen akkor egy egyszerű másol-beilleszt művelettel áttehető a forrás egy forráskód-szerkesztőbe! Ezért képként adjuk meg a forrást!) Figyeljük meg, hogy a „\” jel segítségével elnyomjuk a belső idézőjelek hatását.

Természetesen ez még nem fog működni, hiszen nem írtuk meg a fel- és legördítő algoritmusokat. Azok sem bonyolultak, hát lássuk őket:

```
// legördítő rutin  
function _dSectionExpand(sec) {  
    var a = findObj("vSec"+sec).title;  
  
    if ((a>0) || (findObj("button_go").disabled))  
    {  
        document.all["cSec"+sec].style.display = "none";  
        document.all["xSec"+sec].style.display = "";  
    } else  
        alert("Amíg nem oldottad meg az összes feladatot, addig nem kérheted  
ennek a feladatnak a megoldását!");  
}  
  
// felgördítő rutin  
function _dSectionCollapse(sec) {  
    document.all["xSec"+sec].style.display = "none";  
    document.all["cSec"+sec].style.display = "";  
}  
  
// erre szükségünk lesz, hogy megkeressük az objektumunkat  
function findObj(n, d) {  
    var p,i,x; if(!d) d=document;  
    if((p=n.indexOf("?"))>0&&parent.frames.length) {  
        d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}  
    if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++)  
x=d.forms[i][n];  
    for(i=0;!x&&d.layers&&i<d.layers.length;i++)  
x=findObj(n,d.layers[i].document);  
    if(!x && d.getElementById) x=d.getElementById(n); return x;  
}
```

Látható, hogy igen egyszerűek a rutinok, a leghosszabb rész maga az objektum kikeresése. Figyelmes szemlélő máris észrevehette, hogy van egy elágazás, amely egy figyelmeztető szöveget ír ki, miszerint nem kérheti még a hallgató a feladat megoldását. Erre bizony szükségünk lesz a későbbiekben, de ugye mi előre tervezünk, nem vetjük be magunkat a kódolásba mindenféle előzetes terv nélkül!

Voltaképpen ha ezt a három (négy) Javascript rutint, továbbá a hívást beillesztjük egy HTML fájlba, máris tudunk le- és visszagörgetni. Lássuk az eredményét [3]:

**Gondolkodjunk együtt!**

Most én adok egy forráskódot, a te feladatod az lesz, hogy kilógkázd, hogy fog kinézni a táblázat. Segítségül veheted az előző és az aktuális lapot, vagy ha másképpen nem megy, nézd meg a megoldás és a hozzá kapcsolódó animációt, majd próbáld a feladattal összevetni és megérteni az összefüggéseket.

▲ **Forráskód**

```
<TABLE border="1" width="600" height="300" cellspacing="10">

  <TR bgcolor="#00A0F0" align="center">
    <TD colspan="2" width="50%">A1, A2</TD>
    <TD width="25%">A3</TD>
    <TD width="25%">A4</TD>

  </TR>

  <TR align="center">
    <TD colspan="3">B1, B2, B3</TD>
    <TD rowspan="3">B4, C4, D4</TD>
  </TR>

  <TR align="center">
    <TD colspan="2" width="25%">C1, D1</TD>
    <TD>C2</TD>
    <TD>C3</TD>
  </TR>

  <TR align="center">
    <TD>D2</TD>
    <TD>D3</TD>
  </TR>

</TABLE>
```

▼ **Megoldás**

Mint látható, a Forráskód feliratra kattintva legördül a DIV tartalma, majd újbóli kattintásra eltűnik. Ha a fenti linken kipróbáljuk a Megoldás linket is, megkapjuk a feladat megoldását, persze nem egy egyszerű leírásként, hanem magyarázó szöveggel, animálva, egy Flash fájl formájában. Érdeemes végignézni, a hallgató bármikor megállíthatja, visszatekerheti, újranézheti.

```
<TABLE border="1" width="600" height="300" cellspacing="10">

  <TR bgcolor="#00A0F0" align="center">
    <TD colspan="2" width="50%">A1, A2</TD>
    <TD width="25%">A3</TD>
    <TD width="25%">A4</TD>

  </TR>

  <TR align="center">
    <TD colspan="3">B1, B2, B3</TD>
    <TD rowspan="3">B4, C4, D4</TD>
  </TR>

  <TR align="center">
    <TD colspan="2" width="25%">C1, D1</TD>
    <TD>C2</TD>
    <TD>C3</TD>
  </TR>

  <TR align="center">
    <TD>D2</TD>
    <TD>D3</TD>
  </TR>

</TABLE>
```

1. sor:
kék színű háttér,
vízszintesen középre
igazított cellák

2. sor:
vízszintesen középre
igazított cellák

3 db vízszintesen összekapcsolt cella. Nem adtam
szélességet, mert az előző sorban már
meghatároztam.

Az ilyen jellegű videókat több program is támogat, akár létezik közöttük ingyenes is, érdemes körbenézni a neten, de talán a leghasználhatóbb az Adobe Captivate [4] nevű programja, amely segítségével mindenféle programozási és multimédiás ismeretek nélkül, gyorsan készíthetünk internetes tanulással kapcsolatos, professzionális tartalmakat, gazdag interaktivitással, szoftver- és esetszimulációval, kvizekkel és egyéb látványos megoldásokkal tarkítva. Sajnos, maga a termék nem olcsó, ahogyan az Adobe termékek egyike sem, de ha belegondolunk, milyen lehetőségeket tartalmaz, talán megéri befektetni.

Másik megoldás, hogy elkészítjük az anyagokat Powerpointban (bár ugye az Office csomag sem olcsó!), majd átkonvertáljuk az egészet SWF-be, vagy akár hagyhatjuk PPT (PPS) formában is, a hallgató letölti és levetíti magának.

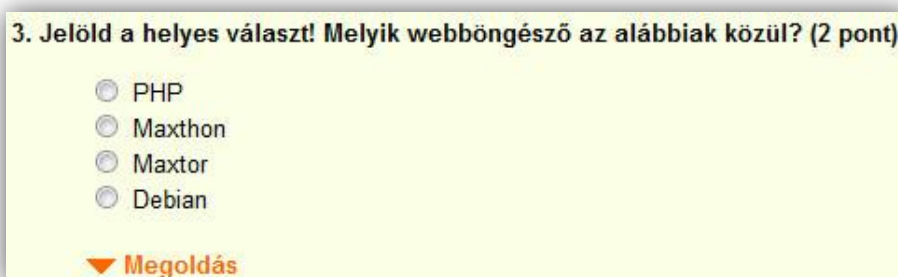
ÖNELLENŐRZÉS

A fenti rutinokat felhasználva könnyen készíthetünk olyan feladatsorokat, amelyek felméri a hallgatót az adott témában. Egy ilyen feladatsor található a képzés végén példaként. Láthatjuk, hogy megpróbál minden fejezetből tudáselemeket bevonni, természetesen ahhoz, hogy minden tudáselem benne legyen, fejezetenként kellene egy-egy feladatsort elkészíteni.

Foglalkozzunk most a technikai megvalósítással, hiszen ha már kész a keret, bármivel fel lehet tölteni azt. Minden feladatnál gondoskodnunk kell arról, hogy meghatározzuk az egyes feladatokra adott pontokat, összegezzük azokat, majd egy meghatározott beosztás szerint értékeljük a hallgatót.

Pontozás: a pontozás kérdése már ott eldőlt, amikor meghatározzuk, milyen típusú feladatokat adunk a hallgatónak a felmérés során. A legelterjedtebbek:

- Egyválasztásos: a több lehetőség közül csak egy a helyes, megvalósítása radio gombokkal történik, a hibás megoldásokra nulla pontot, a helyesre a feladat pontszámát kapja.



- Többválasztásos: több lehetőség, de több bejelölhető opció, Megvalósítása checkbox-okkal történik. A feladatra kapható pontok szétoszlanak a helyes megoldások között, ha nem szeretnénk elárulni, hány helyes megoldás van, akkor ne adjuk meg a

feladatra kapható pontszámot. (Magánvélemény, hogy legyen csak ott a pontszám! Amennyit segít a megoldásban, sokszor annyira hátráltat is, ha a hallgató bizonytalan a tudásában!)

1. Válaszd ki, milyen eszközökre lesz mindenképpen szükséged a weblapszerkesztéshez! (4 pont)

- ☐ Microsoft Word szövegszerkesztő
- ☐ Jegyzettömb
- ☐ Internet
- ☐ Internetböngésző
- ☐ Számítógép
- ☐ Operációs rendszer

▼ **Megoldás**

- Pótlásos feladat: a megadott szavakat, kifejezéseket kell beilleszteni a megfelelő helyre. Minden jó megoldásért jár a pont, az pontszámok összege itt is a feladatra adható összes pontszámmal egyenlő.

2. Párosítsd a hiányos mondatokat a felsorolt kifejezésekkel, szavakkal! (5 pont)

- : az architektúrában ezt az elemet nevezzük kliensnek, segítségével kezdeményezzük az adatforgalmat.
- : az a komponens, amely válaszol a kliens kéréseire.
- : HTML preprocesszor.
- : a szerver által szolgáltatott tartalom.
- : adatbáziskezelő szerver.

Lehetséges válaszok: 1: PHP, 2: MySQL; 3: A weblap forrása; 4: Böngészőprogram; 5: Webszerver

▼ **Megoldás**

- Legördülő lista: voltaképpen hasonló az egyválasztásoshoz. A különbség annyi, hogy több választási lehetőséget tartalmazhat, nem foglalja a helyet.

6. Válaszd ki, melyik tag-gel tudunk félkövér karaktereket készíteni! (2 pont)

Válasz ki a helyes elemet! ▼

▼ **Megoldás**

7. Válaszd ki, melyik tag-gel tudunk villogó karaktereket készíteni! (2 pont)

Válasz ki a helyes elemet! ▼

Válasz ki a helyes elemet!

<I>
<U>
<S>
<SUP>
<SUB>
<BIG>
<SMALL>
<BLINK>
<TT>

Ezekon kívül még ki lehet találni akármilyen változatot, csak a kezeléséről kell gondoskodni. Ha már a kezelésnél tartunk, egy problémával mindenképpen szembesülnünk kell majd. Hogyan adjuk át a helyes megoldásokat és a hozzájuk tartozó pontszámokat a kiértékelő függvénynek? Mivel a hallgató lehet olyan okos, hogy megnézi a forráskódot (hiszen erre tanítjuk!), tegyük valamit ellene, hogy ne legyen olyan egyértelmű a dolga. Sajnos, a használt környezet (HTML, Javascript) kliens-oldali rendszer, ezért alkalmatlan arra, hogy a hallgatótól függetlenül tárolja a megoldásokat. Sebj, tegyük nehezebbé egy picit a dolgát!

Adjuk át a megoldásokat tartalmazó tömböt rejtett mezőként a kiértékelő rutinnak! Fogalmazzuk meg a szabályokat! Az egyválasztásos a legegyszerűbb: ha van négy választási lehetőségünk és abból csak egy a helyes, adjuk neki a pontot, a többi nullázzuk le! A fenti példa alapján („Melyik webböngésző...”) a kiértékelő sor: „00;12;00;00;”. Azaz csak a második megoldásért kap 2 pontot, a többi nullát ér!

A többválasztásos megoldása hasonlóképpen alakul, viszont bele fogunk ütközni egy problémába: ha bejelöli az egészet, akkor megkapja a maximális pontot, ráadásul a feladat összpontszámába is beleszámítana a hibás bejelölés. Ezért a hibás válaszokért nem adunk nulla pontot, jelezzük valamilyen módon, hogy ez nem helyes pipa, ne kumuláljuk az eredményeket. A fenti feladatban a Word és az Internet nem szükséges a fejlesztéshez, a többi 1-1 pontot ér. A megoldókulcs így fog kinézni: „1;-11;-11;11;11;”. Láthatjuk, hogy én mínuszjelet használtam a hibás bejelölés kiszűrésére.

A párosító feladatban nem lesz semmilyen problémánk. Meg kell adjuk sorban a megoldásokat és a pontszámokat. A fenti, párosító feladat alapján a megoldókulcs: „41;51;11;31;21;”, azaz a megoldások rendre: 4, 5, 1, 3, 2, s mindegyik 1-1 pontot ér.

A legördülő lista egyetlen megoldást adhat vissza, kezelését tekintve kicsit más, mint az egyválasztásos. Itt a megoldókulcs (villogó karakterek, azaz blink) a „92” kódsorozat lesz, tehát a 9. válaszért kaphat a hallgató 2 pontot.

Lássuk az egészet kiértékelő függvényt:

```
function check_answers(f)
{
  _jo=new Array();
  _jo[0] = "Gratulálok, ügyes voltál, teljesítetted az elvárt követelményszintet! Biztonsággal mehetsz a következő tanulási egységre és további jó tanulást kívánok! Hiába no, a kitartó tanulás meghozta eredményét!";
  _jo[1] = "Nos, ez ügyes volt! Úgy tűnik, nem volt hiába az a sok foglalkozás a tananyaggal, íme az eredmény. Most már biztonsággal továbbmehetsz a következő tananyagegységre. Sok sikert!";
  _jo[2] = "Gratula, ha nem látom, el sem hiszem! Ügyesen helytálltál, biztonsággal továbbléphetsz a következő tananyagegységre! Ha legközelebb is hasonló eredményt érsz el, teljesen elégedett leszek! Hajrá!";
  _jo[3] = "No látod, nem is volt olyan nehéz! Örülök, hogy sikerrel vetted az akadályokat, bíztam benned. Immár biztonsággal továbbléphetsz a következő tananyagegységre, amihez sok sikert kívánok!";

  _rossz=new Array();
  _rossz[0] = "Sajnos, ez nem sikerült olyan jól, mint kellett volna! Azt ajánlom, térj vissza az előző lapokra, nézd át őket még egyszer, koncentrálj azokra a tananyagegységekre, amelyek most nehezebben mentek! Kitartás, nem eshet minden nap! Ha végképp kimerültél, nyugodtan fordulj konzulensedhez, lehet, hogy csak egy apró (ismeret)lökés hiányzik, s menni fog!";
  _rossz[1] = "Bár \"hát\"-tal nem kezdünk mondatot, de \"Hát ez nem sikerült olyan jól!\". Azért ne vegye el a kedved, inkább próbáld meg újra átnézni azokat a lapokat, amelyek most annyira nem mentek. Ha átnézted, térj ide vissza, s próbáld meg még egyszer elvégezni a feladatokat, mindenféle segítség nélkül. Szerintem jobban fog menni! Ha mégsem sikerülne akkor sem, nyugodtan fordulj konzulensedhez, biztosan segíteni fog! Sok sikert!";
  _rossz[2] = "Hm. Gondolom, te sem ezt vártad...? Azért nem szabad elkeseredned, biztosan csak nem voltál eléggé alapos. Azt ajánlom, térj vissza azokra a tananyagegységekre, amelyek most nem mentek olyan jól, nézd át őket újra. Ha valami nem teljesen tiszta a lapokon, nyugodtan kérdezd konzulensedet a témában, számíthatsz a segítségére. Ne add fel!";
  _rossz[3] = "Ez most nem sikerült, de biztos vagyok benne, nem éred be ennyivel és nem adod fel! Menj vissza az előző oldalakra, szentelj külön figyelmet azokra a tananyagegységekre, amelyek nem sikerültek. Ha valami nem teljesen tiszta, nyugodtan fordulj konzulensedhez, számíthatsz a segítségére! Nem vagy te olyan, aki feladja, pihenj egy picit, majd harcra fel!"

  var obj      = findObj(f);
  var ans      = findObj("answers").value.split(";");
  var bgo      = findObj("button_go");
  var points   = 0;
  var maxpts   = 0;
```

```

for (i=1; i<=obj.length-2; i++)
{
var v = ans[i-1].substr(0,1);
var p = ans[i-1].substr(1,1);
if (p=="-") { p=-1; } else maxpts = maxpts + 1 * p;

if ((obj[i].type=="text") || (obj[i].type=="select-one")) if
(v==obj[i].value) { points = points + 1 * p };
if (obj[i].type=="checkbox") if (v==obj[i].checked) { points =
points + 1 * p; }
if (obj[i].type=="radio") if (v==obj[i].checked) { points = points
+ 1 * p; }
}

var er = Math.floor(points/maxpts*100);

bgo.value="Elért eredmény: "+points+"/"+maxpts+"
("+Math.floor(points/maxpts*100)+"%)";
bgo.disabled=true;
if (er>60) { var l = Math.floor(Math.random()*_jo.length);
alert("Eredményed "+er+"%! "+_jo[l]); } else
{ var l = Math.floor(Math.random()*_rossz.length);
alert("Eredményed "+er+"%! "+_rossz[l]); }
return(er);
}

```

Ne ijedjünk meg a sok szövegtől az elején. Felismerhető a forrásban két tömb, amely a jó és a rossz megoldásokat értékeli. Itt csak két szint létezik: átment, megbukott, de nagyon egyszerűen lehet az elért eredmény függvényében további szinteket definiálni. Ne legyünk lusták, készítsünk minél több reakciót a gép részéről. Dicsérjük meg a hallgatót, ha sikeres volt, adjunk neki pozitív megerősítést, s adjunk szabad utat a következő tananyagegységre. Viszont azt se hagyjuk figyelmen kívül, ha a hallgató nem tudta megfelelő szinten teljesíteni a feladatokat. Ismertessük vele a lehetőségeit, adjunk neki bátorítást az ismétléshez.

Tanulásként nézzük meg a teljes önellenőrzés forráskódját. Mivel mindegyik feladattípusra van benne példa, ettől kezdve már egyértelmű lesz a feladatlapok létrehozása. Amit mindenképpen figyeljünk meg benne:

- a megoldásokat tartalmazó rejtett beviteli mező (answers) felépítése,
- az *insert_span()* függvényben találhatóak a magyarázatok a helyes megoldáshoz,
- nincs megadva az *insert_span()* függvény negyedik paramétere, azaz nem fogja megadni azonnal a megoldásokat, csak ha megnyomja a hallgató a „Megoldottam, ellenőrzést kérek!” gombot.
- a HTML tag-ek „<” és „>” jeleit nem írhatjuk ki, helyettük a < és > kódok használhatóan (less than, greater than)
- ne feledjük növelni az *insert_span()* függvény első paraméterét, azaz a feladat azonosítóját

- a feladatok megoldását a gomb megnyomása a *check_answers('elso')*; kódra aktiválódik, ahol az „elso” annak az űrlapnak a neve, amelyben a feladatok definiálva vannak.

Lássuk a kódot:

```
<head>
<style type="text/css">
@import url(content.css);
</style>
<title> </title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-2"
/>
<script type="text/javascript" src="tip.js"></script>
<script type="text/javascript" src="fa.js"></script>
</head>

<body>

<p class="nodeTitle">Önellenző feladatok</p>

<div class="i_rck">

<span class="iDeviceTitle">Önellenző kérdések és feladatok</span>

<p>
Hajtsd végre az alábbi feladatokat! Ahhoz, hogy biztonsággal
továbbhaladhass, legalább 61%-os szintet kell teljesítened. Figyelj erre
oda, mert a következő részek az itt megtanult tudáselemekre fognak
építeni.<br><br><font color="blue">Jó munkát, a rendelkezésedre álló idő
<b>30 perc</b></font>!
<form name="elso">

<input type="hidden" name="answers" value="1-;11;1-;11;11;11; 41;51;11;31;21;
00;12;00;00; 11;11;11;11; 00;00;00;12; 12; 92; 00;12;00;00; 12;
21;41;31;11;51; 22; 32"><br>

<div class=i_quest>1. Válaszd ki, milyen eszközökre lesz mindenképpen
szükség a weblapszerkesztéshez! (4 pont)</div>
<ul>
<input type="checkbox"> Microsoft Word szövegszerkesztő<br>
<input type="checkbox"> Jegyzetömb<br>
<input type="checkbox"> Internet<br>
<input type="checkbox"> Internetböngésző<br>
<input type="checkbox"> Számítógép<br>
<input type="checkbox"> Operációs rendszer<br>
<script>insert_span(1, "Microsoft Word szövegszerkesztőre nincs
szükségünk, egy egyszerű szövegszerkesztő (pl. Jegyzetömb) tökéletesen
megteszi. Internet sem szükséges, hiszen maga a fejlesztés off-line
történik, tehát egy egyszerű <b>számítógép</b>, <b>operációs
rendszerrel</b>, egy bármilyen <b>böngészőprogrammal</b> és egy
<b>Jegyzetömb</b> már alkalmas a weblapszerkesztés
alapjaira.<br><font color='red'>Ez persze nem jelenti azt, hogy később is
elég lesz!</font>");</script>
</ul>
<br>

<div class=i_quest>2. Párosítsd a hiányos mondatokat a felsorolt
kifejezésekkel, szavakkal! (5 pont)</div>
<ul>
```



```

☐

```

```

<input type=radio name=ugyanaz2>
<!--HTML--><!--HEAD--></HEAD--><!--BODY--></BODY--><!--HTML-->
<br>
<input type=radio name=ugyanaz2>
<!--HTML--><!--HEAD--></HEAD--><!--BODY--></BODY--><!--HTML-->
<br>

<script>insert_span(5, "Természetesen az utolsó megoldás a helyes, hiszen
a <!--HTML--> tag fogja közre a teljes oldalt, a <!--HEAD--> tag-be
írjuk a fejrész definíciókat, majd a <!--BODY--> tag fogja közre a
weblap törzsét.");</script>
</ul>
<br>

<div class=i_quest>6. Válaszd ki, melyik tag-gel tudunk <b>félkövér</b>
karaktereket készíteni! (2 pont)</div>
<ul>
<select>
  <option value="0">Válasz ki a helyes elemet!</option>
  <option value="1"><b>B</b></option>
  <option value="2"><i>I</i></option>
  <option value="3"><u>U</u></option>
  <option value="4"><s>S</s></option>
  <option value="5"><sup>SUP</sup></option>
  <option value="6"><sub>SUB</sub></option>
  <option value="7"><b>BIG</b></option>
  <option value="8"><small>SMALL</small></option>
  <option value="9"><blink>BLINK</blink></option>
  <option value="10"><tt>TT</tt></option>
</select><br>

<script>insert_span(6,"A megoldás a <b>B</b>, ami az angol BOLD
(félkövér) szóból származik!");</script>
</ul>
<br>

<div class=i_quest>7. Válaszd ki, melyik tag-gel tudunk
<blink>villogó</blink> karaktereket készíteni! (2 pont)</div>
<ul>
<select>
  <option value="0">Válasz ki a helyes elemet!</option>
  <option value="1"><b>B</b></option>
  <option value="2"><i>I</i></option>
  <option value="3"><u>U</u></option>
  <option value="4"><s>S</s></option>
  <option value="5"><sup>SUP</sup></option>
  <option value="6"><sub>SUB</sub></option>
  <option value="7"><b>BIG</b></option>
  <option value="8"><small>SMALL</small></option>
  <option value="9"><blink>BLINK</blink></option>
  <option value="10"><tt>TT</tt></option>
</select><br>

<script>insert_span(7,"A megoldás a <blink>BLINK</blink>, ami az angol
\"villogás\" szóból származik!");</script>
</ul>
<br>

<div class=i_quest>8. Melyik a helyes képbeszúrás? (Tételezzük fel, hogy
a silence.jpg ugyanott található, ahol a kód maga! 2 pont)</div>
<ul>
<input type=radio name=ugyanaz3> <br>

```

```

<input type=radio name=ugyanaz3> &lt;IMG src="silence.jpg" width="200"
height="120"&gt;<br>
<input type=radio name=ugyanaz3> &lt;IMG src="kepek/silence.jpg"
width="200" height="120"&gt;<br>
<input type=radio name=ugyanaz3> &lt;IMG src="kepek\silence.jpg"
width="200" height="120"&gt;<br>

<script>insert_span(8, "<ul>1) Az első megoldásban figyeljük meg az
attribútum neveit: <i>with</i> és <i>heigth</i>! Egyik sem helyes... Mi
is a helyes?<br><br>2) A második tökéletesen <b>helyes</b>! Ha ide tette
a jelet, kaptál 2 pontot!<br><br>3) Ez majdnem jó, ha nem jegyeztem volna
meg, hogy a silence.jpg ugyanabban a mappában található, ahol a HTML kód.
Tehát nem adhatok meg útvonalat! Bang! Ez bizony hibás! (Erre mondjuk,
hogy <a href='#' onmouseout='hidetip()' onmouseover='\"tip('Azt a
szabályrendszert jelenti, amely meghatározza, hogy az adott nyelvben
hogyan lehet az egyes nyelvi elemeket (utasításokat) létrehozni,
másképpen <b>helyesírás</b>nak is nevezzük.')]\"
class='\"tips\">szintaktikailag</a> jó, de <a href='#'
onmouseout='hidetip()' onmouseover='\"tip('Logikai, nyelvi helyesség,
működés közben fellépő hibát okozó leírás.<br>(egyszerűen: jó, de nem azt
csinálja, amit szeretnél volna!)]\" class='tips'>szemantikailag</a>
hibás!)<br><br>4) Ugyanaz a hiba, mint az előzőnél, megtéőzve azzal,
hogy nem perjelet, hanem visszaperjelet (backslash-t) használtam! De
csúnya!");</script>
</ul>
<br>

<div class=i_quest>9. Melyik a helyes táblázat struktúraleírás? (2
pont)</div>
<ul>
<select>
  <option value="0">Válasz ki a helyes sorrendet!</option>
  <option
value="1">&lt;TABLE&gt;&lt;TR&gt;&lt;TD&gt;...&lt;/TD&gt;&lt;/TR&gt;&lt;/
TABLE&gt;</option>
  <option
value="2">&lt;TABLE&gt;&lt;TD&gt;&lt;TR&gt;...&lt;/TR&gt;&lt;/TD&gt;&lt;/
TABLE&gt;</option>
  <option
value="3">&lt;TABLE&gt;&lt;ROW&gt;&lt;DATA&gt;...&lt;/DATA&gt;&lt;/ROW&gt;
&lt;/TABLE&gt;</option>
  <option value="4">&lt;TABLE&gt;&lt;TR&gt;&lt;TABLE
DATA&gt;...&lt;/TABLE DATA&gt;&lt;TR&gt;&lt;/TABLE&gt;</option>
</select><br>

<script>insert_span(9,"A helyes megoldás a
&lt;TABLE&gt;&lt;TR&gt;&lt;TD&gt;...&lt;/TD&gt;&lt;/TR&gt;&lt;/TABLE&gt;
struktúra, ezért kaphattál 2 pontot.");</script>
</ul>
<br>

<div class=i_quest>10. Párosítsd a hiányos mondatokat a felsorolt
kifejezésekkel, szavakkal! (5 pont)</div>
<ul>
<input type=text size=1 onkeypress="return handleEnter(this, event)">:
Tábla kezdő TAG<br>
<input type=text size=1 onkeypress="return handleEnter(this, event)">:
Táblában egy sor definíciójára használjuk<br>
<input type=text size=1 onkeypress="return handleEnter(this, event)">:
Egy konkrét cellát határozunk meg vele<br>
<input type=text size=1 onkeypress="return handleEnter(this, event)">:
Egymás mellett levő cellák egyesítése<br>
<input type=text size=1 onkeypress="return handleEnter(this, event)">:
Egymás alatt levő cellák egyesítése<br>

```

```

<br><b>Lehetséges válaszok: </b>1: COLSPAN, 2: TABLE; 3: TD; 4: TR; 5:
ROWSPAN<br>

<script>insert_span(10, "A helyes sorrend az alábbi: TABLE, TR, TD,
COLSPAN, ROWSPAN. Ezért jár az 5 pont, de minden hiba egy pont levonással
jár!");</script>
</ul>
<br>

<div class=i_quest>11. Szeretnék egy FÉRFI/NŐ választási lehetőséget
felkínálni egy űrlapon. Melyik a helyes deklaráció? (2 pont)</div>
<ul>
<select>
  <option value="0">Válassz ki a helyes deklarációt!</option>
  <option value="1">&lt;INPUT name="neme" type="checkbox"
value="ferfi"&gt;FÉRFI&lt;BR /&gt;&lt;INPUT name="neme" type="checkbox"
value="no"&gt;NŐ</option>
  <option value="2">&lt;INPUT name="neme" type="radio"
value="ferfi"&gt;FÉRFI&lt;BR /&gt;&lt;INPUT name="neme" type="radio"
value="no"&gt;NŐ</option>
  <option value="3">&lt;SELECT&gt;FÉRFI&lt;BR
/&gt;NŐ&lt;/SELECT&gt;</option>
</select><br>

<script>insert_span(11,"Bizony, ezek egymást kizáró lehetőségek, tehát a
RADIOgombok készítése volt a helyes megoldás. Tehát a második
lehetőségért kaphattál 2 pontot.");</script>
</ul>
<br>

<div class=i_quest>12. Melyik fogalomra gondoltam? Segítségével a
weblapról kommunikálni tudunk a felhasználóval, elsősorban beviteli célra
alkalmazzuk. (2 pont)</div>
<ul>
<select>
  <option value="0">Válassz ki a helyes fogalmat!</option>
  <option value="1">nyomtató</option>
  <option value="2">mikrofon</option>
  <option value="3">űrlap</option>
  <option value="4">stíluslap</option>
  <option value="5">link</option>
  <option value="6">tábla</option>
</select><br>

<script>insert_span(12,"A helyes megoldás az űrlap!");</script>
</ul>
<br>

<hr color="#80b0ff" width=90% align=left>
<input class="feedbackbutton" type=button value="Megoldottam, ellenőrzést
kérek!" name="button_go" onClick="check_answers('elso');">
</form>

</p>

</body></html>

```


Kiváló funkció lehet és majdnem elkerülhetetlen egy olyan rutin írása, ami a mezőkben nyomott Enter billentyűt nem az űrlap elküldésére használja, hanem átlépteti a következő mezőre a kurzorunkat. Ezeket a beviteli mezőkben kell kezdeményezzük, az alábbi formában: *onkeypress="return handleEnter(this, event)"*, ahogyan az előző forrásban is látszik, például a 2. feladatban. Az ehhez tartozó Javascript forrás a következő:

```
function handleEnter (field, event)
{
    var keyCode = event.keyCode ? event.keyCode : event.which ?
event.which : event.charCode;
    if (keyCode == 13) {
        var i;
        for (i = 0; i < field.form.elements.length; i++)
            if (field == field.form.elements[i])
                break;
        i = (i + 1) % field.form.elements.length;
        field.form.elements[i].focus();
        return false;
    }
    else
        return true;
}
```

FELADATBANK

Végezetül beszéljünk pár mondat erejéig a feladatgyűjteményről, vagy feladatbankról. Célja – ahogy már írtam – olyan feladatok összegyűjtése, amelyek megoldása erősíti a tanultakat, illetve továbbtanulási késztetést vált ki a hallgatóból. Minden esetben adjunk meg egy lehetséges megoldást, s adjunk lehetőséget arra, hogy a hallgató saját megoldásait elküldhesse kiértékelés céljából. Azt vettem észre, hogy néha a hallgatók frappánsabb megoldást adnak, mint amit a tanultak alapján elvárhatnánk tőlük.

A feladatbank semmi más, mint az előzőekben leírtam szintézise. A tárgyalt rutinokat felhasználva készítsünk egy olyan oldalt, ahol feladatokat találhat, megoldásokkal együtt. A megoldásokat engedélyezzük bármikor megtekinteni, bízunk a hallgatóban. Ha az előzőekben sikerült motiválnunk arra, hogy szüksége van az elérhető tudáselemekre, akkor nem fogja előre megnézni a megoldásokat. A feladatbank feladatainak megoldására nincs időkeret adva. Tapasztalatom szerint érdemes mérőföldköveket adni a hallgatók számára. Egyik legjobban bevált módszer azt mondani, hogy csak akkor mehet a következő fejezetre, ha a feladatbankból megoldotta az aktuális fejezet példáit. Természetesen lesz olyan hallgató, aki nem lesz képes megbirkózni a feladatokkal, hiszen nem véletlenül, de vannak benne olyan erősebb feladatok is, amelyek megoldása önálló feldolgozást igényli egy magasabb szintű, de a tanultakkal összefüggő tudásanyagnak. Jelöljük ki a „melegen ajánlott” feladatokat, s legyen az instruktor feladata eldönteni, hogy kitől, milyen megoldási szintet vár el.

 Feladatok **formázás és listák** témakörből

1) Karakterformázás: félkövér, dőlt és aláhúzott

A HTML alapvető formázási műveletei a **félkövér**, *dőlt* és az aláhúzott formázások. Ezeket természetesen lehet keverni, egy karakterrel ***dőlt/félkövér***, de ha nagyon ki akarunk emelni valamit, használhatjuk a ***félkövér/dőlt/aláhúzott*** kombinációt is!

▼ Megoldás

2) Karakterformázás: karakterszínezés

S Z I V Á R V Á N Y

▼ Megoldás

ÖSSZEFOGLALÁS

Mivel minden csoport, a csoporton belül minden hallgató más és más, nem lehet receptet adni sem a távoktatásra, sem a hagyományos oktatásra nézve. Míg a hagyományos oktatásban elsődleges szerepe az oktatónak van, a távoktatás már a tananyagba rejtve próbálja irányítani, s valamilyen szinten uniformizálni a hallgatókat. Hiába mondjuk azt, hogy a távoktatás térben és időben kötetlen, valljuk be, hogy a bizonyos kályha nélkül nehezen tudjuk elkezdni a táncot, s néha-néha becsempészünk a konzultációs időpontokba némi hagyományos oktatást is.

Azon gondolkodtam a dolgozatom elején, hogy vajon helyettesíthető-e a tanár a digitális tananyaggal. Láthattuk a feldolgozás során, hogy egy jól felépített digitális tananyag bizony sok nevelési célt fogalmaz meg. Nem mindegy, hogy milyen célcsoportnak, milyen módszerekkel tanítunk, s hogyan próbáljuk meg őket az általunk elképzelt úton, voltaképpen egyedül elindítani. Adunk-e a kezükbe olyan tananyagot, amely inkább motiválja őket a továbbhaladásban, vagy rosszul terveztük meg a szintjüket, aminek az lesz a következménye, hogy megpróbálnak „túlesni” az anyagon, mindenféle kerülő úton elvégezni a képzést. Nem egyszerű az ő feladatuk sem. Egy megszokott oktatási környezetből kiragadva őket, egy ismeretlen helyzetbe kényszerítjük a hallgatókat, ahol sok esetben a legnagyobb gondot maga a tananyag (és/vagy a keretrendszer) kezelése okozza. Azt kell mondanom, hogy az informatikai írástudás folyamatos fejlődésével ez a gát lassan teljesen megszűnik. Hamarosan elmondhatjuk, hogy az elektronikus oktatás hagyományossá válhat. Ezzel nem azt mondom, hogy felváltja a tanárt, de bizonyos területeken elengedhetetlen lesz a használata. Továbbá – ahogyan a tendencia mutatja – nő az aránya a digitális információszerzésnek a hagyományossal szemben. Bizony, egyre több tudást szerzünk meg digitálisan, mint könyvekből.

Hogy a kérdésre is válaszoljak: igen, a tanár helyettesíthető egy digitális tananyaggal. Csak nem mindegy, hogy milyen az a digitális tananyag. Viszont a digitális tananyag mögött mindig ott lesz az a szakember, aki szakmailag és didaktikailag is elképzei az egész oktatást, analizálja a lehetőségeket, s felépíti azt az optimális tudásanyagot, amely már önállóan is lesz képes megállni a helyét. Magyarán nem, a tanár nem helyettesíthető a digitális tananyaggal. De ki is akarna helyettesíteni minket?

IRODALOMJEGYZÉK

Nemes György, Csilléry Miklós: Kutatás az atipikus tanulási formák (távoktatás / e-learning) modelljeinek kifejlesztésére célcsoportonként, a modellek bevezetésére és alkalmazására

Kutatási zárótanulmány, Budapest, 2006, Nemzeti Felnőttképzési Intézet

Kovács Ilma: Új út az oktatásban? A távoktatás

Budapest, 2005, OKKER Kiadó

Ageless Learner – curious for life! – oktatási portál

<http://agelesslearner.com/>

2010. áprilisi állapot

HIVATKOZÁSOK

[1]: <http://agelesslearner.com/intros/elearning.html>

2010. áprilisi állapot

[2]: <http://unideb.bitwizard.hu/HTML/2.1%20elokeszuletek.html>

[3]: <http://unideb.bitwizard.hu/HTML/5.2%20tablazatok2.html>

[4]: <http://www.adobe.com/products/captivate/>

2010. áprilisi állapot

A Webmester E-learning teljes oktatóanyag megtalálható a

<http://unideb.bitwizard.hu/HTML>

címen.

Az oktatóanyag felhasználható bármilyen szakmai képzés céljára, csupán egy jelzést kérnék a felhasználásról a joco1114@gmail.com email címre.